

AVALIAÇÃO DE DESEMPENHO DO ALGORITMO DE UM PROGRAMA COMERCIAL PARA ROTEIRIZAÇÃO DE VEÍCULOS

CLÁUDIA PELIZARO

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Civil - Transportes.

**ORIENTADOR: Prof. Assoc. Antônio Néilson
Rodrigues da Silva**

São Carlos
2000

AGRADECIMENTOS

Ao professor Antônio Néilson Rodrigues da Silva pela excelente orientação deste trabalho;

À professora Vitória Maria Pureza por suas sugestões e preciosa contribuição para evolução da pesquisa;

Ao professor João Alexandre Widmer, membro da banca examinadora;

Aos meus pais que sempre me incentivaram na realização deste trabalho;

À Fundação de Amparo à Pesquisa do Estado de São Paulo pela bolsa de estudo concedida;

Às empresas que gentilmente contribuíram com informações de grande importância para o desenvolvimento do trabalho;

Aos amigos Cegonha, Bob, Paty e Jú pelo incentivo, carinho e amizade ao longo destes anos;

À GFMI *Software House*.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	ii
LISTA DE QUADROS	iii
LISTA DE ABREVIATURAS E SIGLAS	iv
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS	5
1.2 ABORDAGEM E ESTRUTURA DO TRABALHO.....	5
2 O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS.....	7
2.1 O PRV BÁSICO E SUAS EXTENSÕES	8
2.2 CLASSIFICAÇÃO DOS PROBLEMAS DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS.....	11
2.2.1 Endereços	15
2.2.2 Veículos.....	18
2.2.3 Características do Problema	20
2.2.4 Objetivos	23
3 COMPLEXIDADE COMPUTACIONAL, ESTRATÉGIAS E MÉTODOS DE SOLUÇÕES	26
3.1 COMPLEXIDADE COMPUTACIONAL	26
3.1.1 P e NP.....	32
3.2 ESTRATÉGIAS E MÉTODOS DE SOLUÇÕES	36
3.2.1 Algoritmos Exatos	40
3.2.2 Métodos de aproximação ou heurísticas.....	41
3.3 ANÁLISE DE PERFORMANCE DE HEURÍSTICAS	52
4 SISTEMAS PARA ROTEIRIZAÇÃO DE VEÍCULOS.....	54
4.1 CARACTERÍSTICAS DE SISTEMAS PARA ROTEIRIZAÇÃO.....	56
5 METODOLOGIA	63
5.1 EXPERIMENTAÇÃO EM OTIMIZAÇÃO.....	63
5.2 ESCOLHAS DAS CLASSES DE PROBLEMAS PARA TESTES	66
5.3 ESCOLHA DAS MEDIDAS DE PERFORMANCE.....	67
5.4 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS	69
6 ANÁLISE QUALITATIVA DE SOFTWARE PARA ROTEIRIZAÇÃO DE VEÍCULOS	72
6.1 VISITAS À EMPRESAS	72
6.1.1 Transportadora.....	72
6.1.2 Distribuidora de bebidas.....	79
6.1.3 Distribuidora de produtos alimentícios	86

6.1.4	Conclusão	93
6.2	<i>TRANSCAD</i> VERSÃO 3.2.....	94
6.3	<i>DELIVERY PROFESSIONAL</i>	99
6.4	RESUMO DAS PRINCIPAIS CARACTERÍSTICAS ENCONTRADAS EM TODOS OS <i>SOFTWARE</i> ANALISADOS NESTA SEÇÃO	104
7	ESCOLHA DAS CLASSES DE PROBLEMAS TESTADAS.....	108
7.1	OS QUATORZE PROBLEMAS TESTE DE CHRISTOFIDES, MINGOZZI E TOTH (1979).....	109
7.2	CONJUNTO DE PROBLEMAS DE SOLOMON (1987)	111
7.3	CONJUNTO DE TESTE COM DADOS REAIS	114
7.4	OS TESTES COMPUTACIONAIS	117
7.4.1	Problemas Testes de Christofides, MingoZZi e Toth (1979).....	117
7.4.2	Problemas testes de Solomon (1987)	127
7.4.3	Conjunto de problemas com dados reais	132
8	CONCLUSÕES	144
9	REFERÊNCIAS BIBLIOGRÁFICAS.....	148
	APÊNDICE - ILUSTRAÇÃO DA DISPERSÃO GEOGRÁFICA DOS CLIENTES E DEPÓSITO NOS PROBLEMA TESTES.....	I

LISTA DE FIGURAS

FIGURA 1- UMA INSTÂNCIA DO PROBLEMA DO CAIXEIRO VIAJANTE E A ROTA MÍNIMA PARA ESTE CASO.	27
FIGURA 2 - TENTATIVA DE REPRESENTAÇÃO DO MUNDO NP	35
FIGURA 3- INSERÇÃO DO <i>LINK (i, j)</i>	42
FIGURA 4 - OBJETOS E OPERADORES BÁSICOS.....	54
FIGURA 5 -TELA DO EDITOR DE MAPAS DO <i>TRUCKSTOPS</i>	75
FIGURA 6 - INSERÇÃO DO PV NO <i>SOFTWARE ROADSHOW</i>	82
FIGURA 7 - CADASTRO DOS CLIENTES NO PV NO <i>SOFTWARE ROADSHOW</i>	83
FIGURA 8 - EXEMPLO DE CONSTRUÇÃO DE ROTAS EM ROTEIRIZAÇÃO POR TERRITÓRIO NO <i>SOFTWARE ROADSHOW</i>	85
FIGURA 9 - TELA DO CADASTRO PARA CONSTRUÇÃO DA BASE CARTOGRÁFICA NO <i>SOFTWARE TRUCKS</i> . 88	
FIGURA 10 - CADASTRO DAS FAIXAS DE VELOCIDADES EM FUNÇÃO DA DISTÂNCIA PERCORRIDA, NO <i>SOFTWARE TRUCKS</i>	88
FIGURA 11- ROBUSTEZ DOS <i>SOFTWARE DELIVERY</i> E <i>TRANSCAD</i> EM RELAÇÃO À MELHOR SOLUÇÃO.....	124
FIGURA 12- ROBUSTEZ DOS <i>SOFTWARE DELIVERY</i> E <i>TRANSCAD</i> EM RELAÇÃO À HEURÍSTICA DO SOLOMON.	124
FIGURA 13- GRÁFICO REPRESENTANDO O CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DO NÚMERO DE CLIENTES NO <i>SOFTWARE TRANSCAD</i> , BEM COMO A FUNÇÃO QUE MELHOR REPRESENTA A SUA COMPLEXIDADE COMPUTACIONAL.	126
FIGURA 14 - GRÁFICO REPRESENTANDO O CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DO NÚMERO DE CLIENTES NO <i>SOFTWARE DELIVERY</i> , BEM COMO A FUNÇÃO QUE MELHOR REPRESENTA A SUA COMPLEXIDADE COMPUTACIONAL.	126
FIGURA 15 - REPRESENTAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO, NO <i>SOFTWARE DELIVERY</i> , EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES.....	142
FIGURA 16- REPRESENTAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO, NO <i>SOFTWARE TRANSCAD</i> , EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES.....	142
FIGURA 17. COMPARAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES, NOS <i>SOFTWARE DELIVERY</i> E <i>TRANSCAD</i>	143

LISTA DE TABELAS

TABELA 1: COMPARAÇÃO ENTRE VÁRIAS FUNÇÕES DE COMPLEXIDADE DE TEMPO POLINOMIAL E EXPONENCIAL.....	30
TABELA 2: COMPARAÇÃO DOS VALORES ENCONTRADOS POR 14 HEURÍSTICAS ALTERNATIVAS.	51
TABELA 3: ALGUNS SISTEMAS PARA ROTEIRIZAÇÃO QUE SÃO SIG E SUAS PRINCIPAIS CARACTERÍSTICA	59
TABELA 4: <i>SOFTWARE</i> X CARACTERÍSTICAS GERAIS PARA ROTEAMENTO.....	105
TABELA 5: CARACTERÍSTICAS DOS PROBLEMAS TESTES DE CHRISTOFIDES <i>ET AL.</i> , 1979.....	110
TABELA 6: COMPARAÇÃO DE HEURÍSTICAS APLICADAS SOBRE O CONJUNTO DE DADOS DO SOLOMON.	113
TABELA 7: SOLUÇÕES APRESENTADAS PELO <i>SOFTWARE TRANS CAD</i> PARA OS PROBLEMAS PROPOSTOS POR CHRISTOFIDES <i>ET AL.</i> (1979).	118
TABELA 8: SOLUÇÕES APRESENTADAS PELO <i>SOFTWARE DELIVERY</i> PARA OS PROBLEMAS PROPOSTOS POR CHRISTOFIDES <i>ET AL.</i> (1979).....	119
TABELA 9: COMPARAÇÃO DAS SOLUÇÕES APRESENTADAS PELAS HEURÍSTICAS NOS 14 PROBLEMAS TESTES DE CHRISTOFIDES <i>ET AL.</i> , 1979. ($RAZÃO(i)=DISTÂNCIA/LIMITE INFERIOR(i)$).....	120
TABELA 10: PORCENTAGENS DESVIADAS DAS SOLUÇÕES ENCONTRADAS.....	121
TABELA 11: HEURÍSTICA DO <i>DELIVERY</i> X HEURÍSTICA DO <i>TRANS CAD</i>	121
TABELA 12: COMPARAÇÃO DA PERFORMANCE DAS HEURÍSTICAS.	122
TABELA 13: COMPARAÇÃO GERAL DO DESEMPENHO DOS <i>SOFTWARE DELIVERY</i> E <i>TRANS CAD</i>	123
TABELA 14: ANÁLISE DO TEMPO COMPUTACIONAL DOS <i>SOFTWARE DELIVERY</i> E <i>TRANS CAD</i>	125
TABELA 15: SOLUÇÕES APRESENTADAS PELO <i>SOFTWARE TRANS CAD</i> PARA OS PROBLEMAS PROPOSTOS POR SOLOMON (1987).	128
TABELA 16: SOLUÇÕES APRESENTADAS PELO <i>SOFTWARE DELIVERY</i> PARA OS PROBLEMAS PROPOSTOS POR SOLOMON (1987).	129
TABELA 17: COMPARAÇÃO ENTRE OS <i>SOFTWARE DELIVERY</i> E <i>TRANS CAD</i>	130
TABELA 18: SOLUÇÕES APRESENTADAS PELOS <i>SOFTWARE DELIVERY</i> E <i>TRANS CAD</i> PARA O CONJUNTO DE PROBLEMAS COM DADOS REAIS.	133
TABELA 19: COMPARAÇÃO DO DESEMPENHO DOS <i>SOFTWARE DELIVERY</i> E <i>TRANS CAD</i> EM UMA SITUAÇÃO REAL.	134

LISTA DE QUADROS

QUADRO 1: PRINCIPAIS CARACTERÍSTICAS DE <i>SOFTWARE</i> COMERCIAIS	56
QUADRO 2: SÍNTESE DOS PRINCIPAIS REQUISITOS E CARACTERÍSTICAS DE SISTEMAS PARA ROTEIRIZAÇÃO DE VEÍCULOS SEGUNDO ALGUNS AUTORES.	57
QUADRO 3: BENEFÍCIOS ALCANÇADOS PELA EMPRESA COM A UTILIZAÇÃO DO <i>SOFTWARE TRUCKSTOPS</i>	78
QUADRO 4: CADASTRO DAS PARADAS NO <i>SOFTWARE TRANSCAD</i>	96
QUADRO 5: CAMPOS OPCIONAIS NO CADASTRO DE PARADAS DO <i>SOFTWARE TRANSCAD</i>	96
QUADRO 6: CADASTRO DE VEÍCULOS NO <i>SOFTWARE TRANSCAD</i>	97
QUADRO 7: CARACTERÍSTICAS DO <i>SOFTWARE DELIVERY</i>	100
QUADRO 8: CARACTERIZAÇÃO DOS <i>SOFTWARE DELIVERY</i> E <i>TRANSCAD</i> (DESROCHERS <i>ET AL.</i> , 1990).	106
QUADRO 9: FATORES EXPLORADOS EM CADA PROBLEMA	135
QUADRO 10: ANÁLISE DO TEMPO DE PROCESSAMENTO NO <i>SOFTWARE DELIVERY</i>	138
QUADRO 11: ANÁLISE DO TEMPO DE PROCESSAMENTO NO <i>SOFTWARE TRANSCAD</i>	138
QUADRO 12: RESUMO DOS RESULTADOS PARA O <i>SOFTWARE DELIVERY</i>	139
QUADRO 13: COEFICIENTES DAS VARIÁVEIS E SUA SIGNIFICÂNCIA ESTATÍSTICA (<i>SOFTWARE DELIVERY</i>)	139
QUADRO 14: RESÍDUOS DA REGRESSÃO LINEAR MÚLTIPLA PARA O <i>SOFTWARE DELIVERY</i>	140
QUADRO 15: RESUMO DOS RESULTADOS PARA O <i>SOFTWARE TRANSCAD</i>	141
QUADRO 16: COEFICIENTES DAS VARIÁVEIS E SUA SIGNIFICÂNCIA ESTATÍSTICA (<i>SOFTWARE TRANSCAD</i>)	141
QUADRO 17: RESÍDUOS DA REGRESSÃO LINEAR MÚLTIPLA PARA O <i>SOFTWARE DELIVERY</i>	141

LISTA DE ABREVIATURAS E SIGLAS

PNB	Produto Nacional Bruto
PRV	Problema de Roteirização de Veículos
PRVJT	Problema de Roteirização de Veículos com Janela de Tempo
JT	Janela de Tempo
PV	Ponto de Venda
NP	Não Polinomial
P	Polinomial
SA	<i>Simulated Annealing</i>
AG	Algoritmos Genéticos
BT	Busca Tabu
SIG	Sistemas de Informações Geográficas
IBGE	Instituto Brasileiro de Geografia e Estatística
DER	Departamento de Estradas de Rodagem
PCV	Problema do Caixeiro Viajante
CW	Clarke e Wright (1964) <i>savings algorithm</i>
GM	Gillett e Miller (1974) <i>SWEEP algorithm</i>
MJ	Mole e Jameson (1976) <i>generalized savings algorithm</i>
CMT1	Christofides, Mingozzi e Toth (1979) <i>two-phase algorithm</i>
CMT2	Christofides, Mingozzi e Toth (1979) <i>two-phase algorithm</i>
FJ	Fisher e Jaikumar (1981) <i>two-phase algorithm</i>
DV	Desrochers e Verhoog (1989) <i>MBSA algorithm</i>
AG'	Altinkemer e Gavish (1991) <i>PSA-T algorithm</i>
PF	Pureza e França (1991) <i>tabu search algorithm</i>
T	Taillard (1992) <i>tabu search algorithm</i>
OTS	Osman (1993) <i>tabu search algorithm</i>
OSA	Osman (1993) <i>simulated annealing algorithm</i> ;
GHL	Gendreau, Hertz e Laporte (1994) <i>tabu search heuristic</i> ;
KX	Kelly e Xu (1999). <i>A Set-Partitioning-Based Heuristic for the VRP.</i>

RESUMO

Este trabalho teve como objetivo a avaliação de um *software* comercial de roteirização de veículos. Tal *software*, o *Delivery*, se propõe a ser uma ferramenta de apoio à decisão na escolha da rotina operacional de coleta e/ou distribuição física de produtos, através da criação de roteiros alternativos, o que possibilita analisar a viabilidade de implantação da rotina operacional. A proposta original consistia em desenvolver uma metodologia para testar e avaliar a qualidade das soluções geradas pelo algoritmo deste sistema. O trabalho foi conduzido através de uma pesquisa bibliográfica dos problemas clássicos de roteirização e programação de veículos, abordando suas classificações, estratégias e técnicas de solução. Um estudo em empresas que utilizam procedimentos sistemáticos de roteirização foi realizado, com a intenção de caracterizar o cenário em que se desenvolve a atividade de distribuição física. Neste estudo foi possível identificar as características mais relevantes para sistemas comerciais de roteirização de veículos, bem como caracterizar os *software* utilizados pelas empresas em questão. Finalmente, realizou-se uma análise empírica comparativa entre os *software Delivery* e *TransCAD* através da aplicação de problemas testes encontrados na literatura que representam algumas classes do problema de roteirização de veículos, além da aplicação de um caso real. Resultados demonstraram que a heurística do *software TransCAD* apresenta melhor desempenho que a do *software Delivery*.

Palavras-chave: sistemas comerciais para roteirização de veículos; performance de heurísticas; análise empírica de heurísticas.

ABSTRACT

The aim of this work is to evaluate a commercial computer program for vehicle routing. The software, named *Delivery*, has been designed to be a decision-support tool for planning goods collection and/or distribution. Its capacity for creating several alternative routes is very useful in the analysis of possible operational schemes before their actual implementation. A methodology for testing and evaluating the quality of the solution generated by the algorithm has been applied in this work, after a comprehensive literature review of the traditional vehicle routing and scheduling problems, their classification, and solution techniques and strategies. A field study in some companies that actually use a similar tool for routing their fleets has been carried out, in order to better understand how the activity is performed in real world conditions. The most important characteristics of commercial vehicle routing systems has been also identified in the field study, as well as the software used by the studied companies. Finally, a comparative empirical analysis with the software *Delivery* and *TransCAD* has been carried out. In order to compare them, test problems available in the literature, that correspond to some of the most common vehicle routing problems, and a real case application were employed. The results have shown that the heuristic of *TransCAD* had a better performance than the one used in *Delivery*.

Keywords: commercially vehicle routing software; heuristics performance; empirical analysis of heuristics

1 INTRODUÇÃO

Segundo Eilon *et al.* (1971), logística pode ser definida como “a provisão de bens e serviços de um ponto de oferta para um ponto de demanda”. Um completo sistema logístico abrange o processo de movimentação de matéria-prima (e outros insumos necessários à produção) de fornecedores para a fábrica, a conversão desses insumos em produtos na fábrica, o movimento destes produtos para vários armazéns ou depósitos, e a eventual entrega destes produtos ao consumidor final.

Para Ballou (1993) a logística associa estudo e administração dos fluxos de bens e serviços e da informação que os põe em movimento. Caso fosse viável produzir todos os bens e serviços no ponto onde eles são consumidos ou caso as pessoas desejassem viver onde as matérias-primas e a produção se localizam, então a logística seria pouco importante. Mas isto não ocorre. Uma região tende a especializar-se na produção daquilo que tiver vantagem econômica para fazê-lo. Isto cria um hiato de tempo e espaço entre matérias-primas e produção e entre produção e consumo. Vencer tempo e distância na movimentação de bens ou na entrega de serviços de forma eficaz e eficiente é tarefa do profissional de logística. Ou seja, sua missão é colocar as mercadorias ou os serviços certos no lugar e no instante correto e na condição desejada, ao menor custo possível.

A atividade de distribuição de uma empresa compreende toda movimentação e estocagem de bens “a jusante” da fábrica. A última etapa nesta movimentação (dos centros de distribuição para os consumidores), a qual pode ser definida como transporte local ou entrega, representa o elo mais caro da cadeia de distribuição

(Christofides, 1981). Para esta etapa ser realizada de maneira eficiente, a empresa deve desenvolver o planejamento e a execução desta atividade de transporte de forma racional (Bodin *et al.*, 1983).

A importância dos problemas de distribuição diz respeito a magnitude dos custos associados à esta atividade. Estudos realizados por Kearney (1989¹, *apud* Bodin *et al.*, 1983), mostraram que os custos de distribuição física agregam cerca de 16% ao valor final de um item.

Ballou (1993) identifica que os custos logísticos de produtos tangíveis representam cerca de 23% do PNB, nos Estados Unidos, e destes custos, o transporte representa algo em torno de dois terços. O autor afirma ainda que estes percentuais tem-se mantido constante ao longo do tempo.

Um estudo realizado por Read e Muller² (1991) para conceituar a qualidade em logística, consistiu na pesquisa em cerca de 225 empresas americanas e européias de diferentes setores industriais. O resultado foi que mais de 75% dos entrevistados apontam os serviços realizados no prazo entre as três primeiras colocações.

Assim, é evidente a importância dos problemas de distribuição vinculados à roteirização e programação de veículos, pela magnitude dos custos associados à esta atividade e pela qualidade de serviço proporcionado por ela.

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

¹ KEARNEY, A. (1980). *Improving productivity in physical distribution*. Report undertaken for CPDM. London.

² READ, W. F.; MULLER, M. S. The state of quality in logistics. *International Journal of Physical Distribution & Logistics Management*, v.21, n.6, p.32-47.

aos sistemas de distribuição. Concomitantemente, aumentam as exigências dos clientes com relação a prazos, datas e horários de entrega.

Com o aumento das restrições operacionais (e portanto da complexidade do problema), encontrar intuitivamente uma “boa” solução na programação e execução das operações diárias de transporte e distribuição física se torna cada vez mais difícil. É neste nível que a necessidade de uma ferramenta flexível e eficiente no planejamento da atividade de roteirização e programação se mostra evidente.

Segundo Solomon e Desrosiers (1988), a roteirização e programação de veículos é um importante segmento de muitos sistemas de custo de transporte e distribuição. O problema de roteirização de veículos (PRV) determinístico envolve o planejamento de um conjunto de rotas de mínimo custo, que inicia e termina em pelo menos um depósito central, através de uma frota de veículos que atende a um conjunto de clientes com demanda conhecida.

Quando o aspecto espacial do problema de roteirização de veículos é combinado com o aspecto temporal, tem-se o problema de roteirização de veículos com janela de tempo (PRVJT). Neste problema, deve ser levado em consideração o horário de entrega, ou janela de tempo, derivado do fato de que alguns clientes impõem prazos de entrega e horários de atendimento.

A roteirização de veículos tem sido vista como um dos maiores sucessos na área de Pesquisa Operacional nas últimas décadas. Este sucesso pode ser associado à atuação conjunta de teoria e prática. Por um lado a pesquisa operacional, como filiação acadêmica, tem desenvolvido algoritmos que têm um importante papel na implementação de sistemas de roteirização. Por outro lado, o desenvolvimento de *hardware* e *software* e sua crescente integração na atividade operacional com foco comercial, tem criado um alto grau de percepção dos benefícios potenciais da roteirização de veículos, criando um interesse generalizado por parte das empresas usuárias.

É neste cenário que a empresa GFMI *Software*, está desenvolvendo um sistema comercial de roteirização de veículo - o *Delivery*. O *software Delivery* se propõe a ser uma ferramenta de apoio à decisão na escolha da rotina operacional de coleta e/ou distribuição física de produtos, através da criação e simulação de roteiros alternativos, possibilitando desta forma analisar a viabilidade de implantação da rotina operacional. O programa, composto por uma base cartográfica com a localização de todas as cidades do Brasil e por um banco de dados integrado a esta base cartográfica (embora não se trate de um SIG), permite que o usuário entre com várias restrições operacionais, tais como janelas de tempo, variação da velocidade do veículo por trecho ou tipo de rodovia, prioridade de atendimento, tempo de parada e prazo de entrega. Ao final do processamento, o usuário visualiza os resultados em mapas e sob a forma de relatórios.

Uma questão fundamental no desenvolvimento de sistemas desta natureza diz respeito à interação entre o despachante e o computador. A divisão da responsabilidade no roteamento entre o despachante e o sistema varia para diferentes aplicações e depende: da qualidade da solução desejada; da habilidade do algoritmo de gerar soluções de alta qualidade, associado ao tempo computacional; da quantidade de tempo que o despachante tem e leva para fazer construções interativamente e modificar soluções; e da estabilidade do ambiente operacional.

No entanto, a qualidade das soluções apresentadas deve ser no mínimo, aceitável. Identificações repetidas de erros e ineficiências podem fazer com que o despachante perca a confiança nas soluções apresentadas pelo sistema. Assim, a “qualidade da solução” deve ser considerada em todas as fases do desenvolvimento do sistema, formulação e solução do modelo, e implementação do sistema. É com base nos elementos observados acima que são definidos os objetivos deste trabalho, apresentados a seguir.

1.1 OBJETIVOS

Principal:

- Desenvolver uma metodologia para avaliação quantitativa do desempenho demonstrado pelo algoritmo de um *software* comercial para roteirização de veículos.

Secundários:

- Definir, caracterizar e classificar problemas de roteirização, estratégias e métodos de solução, a partir de alguns casos clássicos;
- Realizar um estudo em empresas comprometidas com roteirização de veículos, para levantamento das principais características levadas em consideração na formulação de seus problemas e caracterização dos *software* por ela utilizados para dar suporte na realização da atividade.

1.2 ABORDAGEM E ESTRUTURA DO TRABALHO

O trabalho foi conduzido por uma ampla pesquisa bibliográfica dos problemas clássicos de roteirização de veículos, abordando a definição do problema, suas classificações, estratégias e técnicas de solução, bem como a complexidade computacional envolvida.

Sistemas comerciais para roteirização de veículos também foram tema de pesquisa, buscando-se informações com relação aos *software* disponíveis no mercado, por quem são utilizados e de que forma, tendo como finalidade identificar as características que são relevantes em sistemas desta natureza. Para tanto foi feito um estudo em empresas engajadas com o problema de roteirização, e que fazem uso de alguma ferramenta computacional para auxiliar nesta atividade, identificando o problema prático (real) e como ele é formulado (problema teórico) para identificação das características que são relevantes. A identificação de tais características contribuiu no processo de escolha das classes de problemas em roteirização de veículos que foram testadas e os principais fatores que foram explorados para análise de desempenho do *software*, objetivo principal do trabalho.

Por fim, foi realizada a avaliação de desempenho do algoritmo do *software* comercial *Delivery*, valendo-se de uma análise empírica comparativa da performance de heurísticas, tendo por base problemas padronizados encontrados na literatura, criados exatamente para fins de teste, além de um problema real de roteirização. O *software* comercial *TransCAD* foi também utilizado para auxiliar neste procedimento de análise empírica. Conclusões foram tiradas comparando-se os valores encontrados neste estudo empírico, tomando como referência os resultados encontrados na literatura.

O presente trabalho está organizado da seguinte forma: no Capítulo 2 é definido o problema de roteirização de veículos, apontando suas variações e apresentando um esquema de classificação proposto na literatura para tratá-las. O Capítulo 3 aborda a complexidade computacional em função das estratégias e métodos empregados para solução do problema de roteirização de veículos. Aqui, ainda foi mostrada a necessidade e as estratégias para se analisar a performance dos algoritmos empregados na solução de tais problemas. O Capítulo 4 fala sobre sistemas para roteirização de veículos, enquanto o Capítulo 5 apresenta a metodologia seguida no desenvolvimento deste estudo. O Capítulo 6 trata dos aspectos qualitativos de *software* para roteirização disponíveis no mercado. Este capítulo é especialmente importante no sentido de apontar os fatores e características a serem explorados durante o processo de escolha das classes de problemas relevantes de serem testadas. O Capítulo 7 traz as classes de problemas escolhidas como objeto de análise, o conjunto de problemas testes que irão representá-las, e sua aplicação, bem como a apresentação e análise dos resultados. As conclusões são apresentadas no Capítulo 8, seguidas das referências bibliográficas.

2 O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS

Christofides (1985) considera o problema de distribuição como sendo aquele em que os veículos, localizados em um depósito central são requisitados para visitar – durante um dado período de tempo – clientes geograficamente dispersos, para cumprir suas exigências. Este problema aparece em um grande número de situações práticas, relativa a distribuição de mercadorias e é conhecido pelo nome genérico de *Problema de Roteirização de Veículos* (PRV). O PRV é também conhecido na literatura como “Programação de veículos” (Clarke & Wright, 1964; Gaskell, 1967), “Despacho de Caminhões” – “*Trucks Dispatching*” - (Dantzig & Ramser, 1959³; Christofides & Eilon, 1969⁴; Krolak *et al.*, 1972⁵), ou simplesmente “Problemas de Entrega” (Hays, 1967⁶). Frequentemente, eles aparecem também em situações que não estão relacionadas com a entrega de mercadorias. Por exemplo, a coleta de cartas de caixas de correio ou fichas de telefones públicos, roteirização para serviços de atendimento médico domiciliar, roteirização de carteiros, etc.

³ DANTZING, G. B.; RAMSER, J. H. (1959). The truck dispatching problem. *Management Science*, v.6, pp. 80-91.

⁴ CHRISTOFIDES, N.; EILON, S. (1969). An algorithm for the vehicle dispatching problem. *Operational Research*, n. 20, pp. 309-318.

⁵ KROLAK, P.; FELTS, W.; MARBLE, G. (1972). A man-machine approach toward solving the traveling salesman problem. *Transportation Science*, n.6, p. 146-170.

⁶ HAYS, R. (1967). *The delivery problem*. Department of Management Science, Carnegie Institute of Technology, Pittsburgh, 10PA. [12:1] (Report 6).

Existem muitas variações do PRV nas quais as operações podem ser de coletas e/ou entregas e as “exigências dos clientes” e os “veículos” podem assumir muitas variedades.

Tendo em vista o grande número de situações práticas que dão origem aos PRV's, Christofides (1985) apresenta o Problema *básico* de Roteirização de Veículos, e o descreve como a essência de *todos* os Problemas de Roteirização de Veículos.

O PRV básico roteiriza os veículos (uma rota por veículo, começando e terminando no mesmo depósito), de forma que todos os clientes são atendidos em suas demandas e o custo total da viagem é minimizado.

2.1 O PRV BÁSICO E SUAS EXTENSÕES

O problema básico de roteirização ignora um grande número e variedade de restrições adicionais e extensões, que são freqüentemente encontradas em problemas ou situações reais. Algumas destas restrições e extensões são listadas por Christofides *et al.* (1982⁷, *apud* Christofides, 1985):

- (i) Cada veículo pode ser operado em mais de uma rota, desde que o tempo total gasto por estas rotas seja menor que um dado período T (que é relativo ao período de operação do sistema). Note que tal limitação requer o conhecimento do tempo de viagem (t_{ij}) entre cada par de clientes.
- (ii) Cada cliente deve ser visitado durante seu horário de funcionamento ou em um determinado período compreendido em uma *janela de tempo*.
- (iii) O problema pode envolver tanto entregas como coletas de clientes. Adicionalmente, é possível misturar entregas e coletas em uma única

⁷ CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. (1982). *MOVER (Modeling and Optimisation of Vehicle Routing) – A User Manual*, Imperial College of Science and Technology, London. [12:3]

rota, ou alternativamente, pode ser exigido que o veículo execute primeiro todas as entregas na rota antes das coletas. Este último caso é geralmente conhecido como *backhauling*.

- (iv) Assim como em (ii), onde cada cliente está relacionado a um horário de funcionamento ou janela de tempo, os veículos (na realidade seus motoristas) podem ter também uma janela de tempo de trabalho durante o período. Assim, o veículo só poderá ser operado durante a janela de tempo especificada.
- (v) O tempo consumido para realizar a atividade também deve ser considerado. Isto inclui: tempo de descarga (ou tempo de carregamento, no caso de coletas) em cada cliente; tempo de carregamento do veículo no depósito – para a primeira viagem e para as subseqüentes (ver (i)); tempo de fila do veículo para ser carregado no depósito se o número de baias disponíveis é limitado; etc.

Apesar das limitações e extensões listadas acima serem apenas uma pequena fração daquelas encontradas na prática, elas – e muitas outras – não mudam a natureza essencial do PRV básico e podem ser incorporadas em um grande número de modelos heurísticos para resolver o problema.

Por outro lado, existem algumas outras considerações práticas que também ocorrem freqüentemente, que não se ajustam adequadamente dentro da ferramenta básica para PRV. Algumas delas são:

Múltiplos depósitos - empresas com mais de um depósito, onde estes operam de forma dependente, ou seja, o veículo pode sair de um depósito e depois de visitar os clientes retorna a outro depósito, podendo ser carregado novamente e continuar em uma viagem subseqüente. Neste caso, os depósitos não podem ser considerados isoladamente. Quando o depósito é autônomo, ou seja, cada um tem sua própria frota de veículos e sua própria área de cobertura geográfica para atendimento dos clientes, o problema deve ser simplificado em vários problemas similares de roteirização de veículos com um único depósito.

Nível de serviço ao consumidor – o nível de serviço é medido pelo período de tempo durante o qual as exigências dos clientes são cumpridas. Como os clientes e seus pedidos consistem em um processo dinâmico e não periódico, qualquer tentativa para definir o problema de roteirização de veículos para um dado período deve ser, por definição, uma aproximação ou uma arbitrariedade imposta. Algumas destas aproximações são:

- (a) *Tipicamente periódico*: é o caso em que os clientes são fixos e os seus pedidos são esperados a cada t dias. Assim, o cliente deve ser visitado T/t vezes durante um período de T dias, e estas visitas devem acontecer a cada $t \pm \epsilon$ dias, para algum valor pequeno de ϵ . As rotas fixas geradas na solução do problema de roteirização de veículos para o período são executadas de forma tão freqüente que cada cliente sabe quando esperar por suas entregas.
- (b) *Prazo fixo*: um modo de operação freqüentemente utilizado é fixar um prazo na data de entrega dos pedidos. Os pedidos recebidos nos primeiros T dias serão entregues nos próximos T dias (sendo T um período de tempo a ser especificado). Assim, os pedidos recebidos no período corrente serão ignorados do problema de roteirização deste mesmo período, fazendo parte do problema somente no próximo período.
- (c) *Prioridades de clientes*: uma forma alternativa para definir um período para atender os clientes, assim como em (a) e (b), é alocar uma prioridade para cada cliente, de acordo com o intervalo de tempo restante para a data em que ele deve ser visitado. Quanto menor o tempo restante, maior será a prioridade. Este tipo de problema tem um objetivo complexo que envolve tanto custo de roteirização, quanto as prioridades dos clientes roteirizados, em tentativa de manter o serviço com um atraso máximo de T -dias.

Múltiplas mercadorias – em alguns problemas de roteirização de veículos, os veículos são compartimentados de forma que diferentes mercadorias são armazenadas em compartimentos segregados. Cada cliente pode requerer quantidades específicas de diferentes tipos de mercadorias. Tais problemas aparecem na distribuição de combustível, comidas (congeladas ou não), etc., e envolve – além

do problema de roteirização de veículos – aspectos do “problema da mochila”. Neste último, um conjunto de n itens está disponível para ser empacotado em uma mochila de capacidade C unidades. Cada item i tem valor vi e utiliza ci unidades da capacidade. Deve-se determinar o subconjunto I de itens a ser colocado dentro da mochila de maneira a maximizar a somatória de vi ($i \in I$) – pois quanto maior o valor da mercadoria maior será o lucro com o frete - tal que a soma de ci ($i \in I$) seja menor que C (Reeves,1993).

Objetivos múltiplos e conflitantes – em algumas ocasiões podem surgir situações em que o problema de roteirização de veículos seja simplesmente impossível de ser solucionado (em algumas situações em que se pretende cumprir o nível de serviço oferecido, mas o sistema não comporta um dado pico de demanda, por exemplo). Na prática esta inviabilidade é resolvida contratando-se veículos extras e/ou adiando o serviço para alguns clientes. Nestas circunstâncias, o objetivo pode ser minimizar:

- (1) o número de veículos extras contratados; e/ou
- (2) o número (ou a soma de cargas) de clientes não servidos no período presente; e/ou
- (3) a distância total (ou tempo) viajada.

Em geral, o objetivo deve ser uma combinação linear destes três itens.

2.2 CLASSIFICAÇÃO DOS PROBLEMAS DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS

A saída básica de todos os sistemas para roteirização e programação é essencialmente a mesma: para cada veículo ou motorista, uma rota e uma programação é providenciada. Geralmente, a rota especifica a seqüência dos locais a serem visitados e a programação identifica o horário em que vai ocorrer a atividade nestes locais.

Encontram-se na literatura vários esquemas para classificação dos problemas de roteirização de veículos. Bodin *et al.* (1983) propõem uma estrutura que classifica

os problemas em função de suas restrições de aspectos espaciais e/ou temporais. Assim, os problemas são divididos em três grandes grupos: (1) problemas de roteirização pura; (2) problemas de programação; e (3) problemas de roteirização e programação. Depois estes grupos são subdivididos de acordo com um esquema mais detalhado.

Assad (1988) alega que a maior dificuldade em encontrar um esquema de classificação apropriado para problemas de roteirização, está em “o que” tomar como base para classificá-los: os requisitos do problema ou a técnica de solução proposta. O autor sugere que cada problema prático seja caracterizado particularmente, de acordo com um conjunto de elementos por ele listado. Outra possível classificação baseia-se no tempo em que as informações das demandas estão disponíveis. Nos problemas clássicos de roteirização, assume-se que a demanda é conhecida antecipadamente (demanda determinística). Na roteirização dinâmica, a demanda é estocástica, ou seja, ocorre em tempo real e é inserida no roteamento em andamento.

Desrochers *et al.* (1990) propuseram um esquema muito elegante, que além de servir para classificar uma grande variedade de problemas de roteirização e programação de veículos, dá suporte ao desenvolvimento de modelos para sistemas desta área. A idéia é dar diretrizes para uma representação teórica do problema real, servindo como base para o desenvolvimento de modelos e sistemas, e possibilitar uma escolha apropriada do algoritmo que irá tratá-lo.

Este esquema será descrito mais detalhadamente, pois servirá como ferramenta para caracterização dos *software Delivery* e *TransCAD*, proporcionando uma direção para o processo de escolha das classes de problemas a serem neles testadas.

Tal esquema manipula informações em três níveis diferentes. No primeiro nível, está o *problema situado na vida real*. Ele pode conter muitos aspectos que não são relevantes para a seleção de um método de solução. No segundo nível está o *problema tipo teórico*, obtido do problema situado na vida real por determinação e

modelagem das entidades relevantes que são descritas em termos de decisões, objetivos e restrições. No terceiro nível estão os *algoritmos*.

Inicialmente, é proposta uma linguagem para definir o *problema teórico* em roteirização e programação de veículos. A linguagem proposta faz uso de quatro campos (*fields*).

O primeiro campo descreve as características e restrições que são relevantes somente para um único endereço (clientes e depósitos). É preferido o termo “endereço” a “clientes” devido à grande variedade de tipos de clientes: os usuais clientes de um único endereço; clientes correspondentes a um par origem-destino; ou clientes que são definidos como sendo todos os endereços localizados em um segmento de rua.

O segundo campo especifica as características relevantes somente para um único veículo. O terceiro campo contém todas as características do problema que não podem ser identificadas com os endereços ou veículos. O quarto campo define uma ou mais funções objetivos.

Um quinto campo pode ser adicionado para descrever informações adicionais sobre uma instância específica da classe do problema. Embora tais informações não façam parte do modelo definido pelos quatro campos, elas podem ser úteis para a seleção de um algoritmo adequado. Por exemplo, ele pode ajudar a saber o número médio de clientes que serão atendidos por um veículo. A especificação deste campo pode ser adiada até o desenvolvimento do sistema em um estágio mais avançado.

O esquema de classificação é projetado para modelar a situação problema real e para descrevê-la em uma forma padrão, de maneira que exista uma correspondência entre as características do problema real e os componentes do esquema. O esquema lida apenas com problemas estáticos, onde os dados não mudam durante o planejamento de uma sessão. No entanto, ele permite que alguns dados sejam estocásticos.

Há duas suposições básicas que o usuário deve estar atento. Primeiro, localização e distâncias são definidas em termos de uma dada rede, e informações sobre barreiras naturais são incorporadas nesta rede. Segundo, a atividade de distribuição deve ser planejada para um único dia, exceto no caso de roteamento periódico (esta distinção será abordada no item 2.2.1).

Existem casos em que alguns aspectos da modelagem exigem a imaginação do usuário. Considere, por exemplo, uma situação em que o motorista tem um período mínimo de descanso entre dois dias de trabalho. Isto pode ser modelado como um problema de roteirização de veículos com janela de tempo não-idêntica para o veículo: uma vez que o fim da jornada de trabalho é conhecida, é possível determinar a disponibilidade dos motoristas em função das janelas de tempo dos veículos.

O esquema de classificação também permite a modelagem de restrições flexíveis, no sentido de função penalidade. Como um exemplo, considere a situação em que cada motorista pode cumprir um certo limite de horas extras de trabalho. Isto pode ser modelado pela combinação de restrições rígidas e flexíveis na duração da rota: as restrições rígidas impõem um limite sobre as horas extras, e as restrições flexíveis penalizam horas extras.

Todos os elementos na definição do problema são, em princípio, unidimensionais. No entanto, um sobrescrito pode ser adicionado para indicar restrições multidimensionais. Por exemplo, cap_i indica que a capacidade do veículo é limitada em uma dimensão, que pode ser, por exemplo, peso ou volume; cap_i^2 indica restrição de capacidade em duas dimensões, que pode significar que os limites tanto de peso como de volume estão sendo levados em conta.

A linguagem de classificação consiste em um conjunto de regras que definem uma estrutura permissível. Cada regra define um símbolo não terminal em termos de outro símbolo não terminal (campo, subcampo e elementos) e símbolo terminal (valor dos elementos ou sinais); o símbolo V é usado para representar um “ou”. Cada

símbolo não terminal é colocado entre colchetes angulares. Cada sinal é seguido por um comentário (interpretação) entre chaves. O símbolo \circ é usado para indicar um valor padrão, que geralmente é o valor mais simples ou o mais freqüente.

Cada classe do problema sobre consideração é definida por um número de símbolos, alguns dos quais podem ser iguais a \circ . Para uma notação conveniente, dois símbolos sucessivos são separados por uma barra vertical se eles pertencerem a campos diferentes e por uma vírgula se pertencerem ao mesmo campo e não forem iguais a \circ .

Note que foi feita a escolha por um resumo verbal para interpretação de cada símbolo ao invés de uma completa definição em termos matemáticos. Assim, o esquema de classificação segue com a seguinte estrutura:

```
<classificação> ::=
    <endereço>
    <veículo>
    <características do problema>
    <objetivos>
```

2.2.1 Endereços

O primeiro campo define as características que podem ser associadas a um único endereço. Os endereços deverão estar localizados sobre uma rede $G = (V,E)$ com um conjunto V de nós e um conjunto E (não direcionado) de arestas e de arcos (direcionado). É composto de quatro subcampos.

O primeiro subcampo especifica o *número de depósitos*. Existem problemas de um único depósito e problemas onde o número de depósitos é dado como parte da instância do problema.

O segundo subcampo especifica o *tipo de demanda*. Existem três partes. A primeira corresponde a localização da demanda: ° indica que o cliente está localizado sobre o nó, ARESTA indica que o cliente está localizado nos arcos da rede, MISTO indica que os clientes estão localizados tanto nos nós como nos arcos, e TAREFA indica que cada cliente está associado a um par origem-destino; a carga é recolhida em um endereço de origem e entregue em um endereço destino. A segunda parte deste subcampo especifica se todas as demandas são do mesmo tipo (só entregas ou só coletas) ou não (mistura de entregas e coletas). A terceira parte especifica a natureza da demanda: determinística ou estocástica.

O terceiro subcampo especifica *restrições de programação dos endereços*, isto é, os aspectos temporais da demanda. Podem não existir restrições temporais, ou o horário de saída é fixo (programação fixa), ou o horário de saída é restrito por um único intervalo (janela de tempo única), ou por um conjunto de intervalos (janela de tempo múltipla).

O último subcampo especifica a *restrição de seleção dos endereços*. Existe uma distinção básica entre duas classes de problemas. Na primeira classe, um planejamento simples é construído para uma dada coleção de endereços e veículos. Há três subclasses: todos os endereços devem ser visitados; um dado subconjunto de endereços deve ser visitado e os outros serão visitados se for vantajoso; ou os endereços são particionados em subconjuntos e pelo menos um endereço em cada subconjunto deve ser visitado. Na segunda classe, um número de planos é construído para um certo período de tempo, durante o qual os endereços devem ser visitados com uma dada prioridade ou com uma dada frequência. Este requisito está aberto para várias especificações. Ele possibilita conduzir problemas de longo horizonte de tempo, no qual um problema de alocação semanal é solucionado antes da definição do problema de roteirização diária.

<endereços>::=

<número de depósitos>

<tipo de demanda>

<restrição de programação de endereços>

<restrição de seleção de endereços>

<número de depósitos> ::= 1 V l

1 [um depósito]

l [especificado como parte da instância do problema]

<tipo de demanda> ::= $\langle \alpha_1 \rangle$ V $\langle \alpha_2 \rangle$ V $\langle \alpha_3 \rangle$

$\langle \alpha_1 \rangle$::= ° V ARESTA V MISTO V TAREFA

° [roteirização dos nós]

ARESTA [roteirização dos arcos]

MISTO [roteirização mista (nós e arcos)]

TAREFA [roteirização tarefa]

$\langle \alpha_2 \rangle$::= ° V \pm

° [só entregas ou só coletas]

\pm [mistura entregas e coletas]

$\langle \alpha_3 \rangle$::= ° V \sim

° [demanda determinística]

\sim [demanda estocástica]

<restrição de programação de endereço> ::= ° V pf_j V jt_j V jm_j

° [sem restrição de horário]

pf_j [programação fixa]

jt_j [janela de tempo simples]

jm_j [janela de tempo múltipla]

<restrição de seleção de endereços> ::= ° V *subconjunto* V *escolha* V *período*

° [planejamento simples; todos endereços devem ser visitados]

subconjunto [planejamento simples; um dado subconjunto de endereços deve ser visitado e os outros serão visitados se for vantajoso]

escolha [planejamento simples; pelo menos um endereço de cada subconjunto particionado deve ser visitado]

período [planejamento construído para um dado período de tempo]

2.2.2 Veículos

O segundo campo define as características do veículo em suas rotas. Há três tipos de informações neste campo: o número de veículos, a característica física do veículo, e as restrições temporais sobre uma rota.

O primeiro subcampo especifica o *número de veículos*: o número de veículos é uma constante, especificado como parte de um tipo de problema, ou uma variável, especificado como parte da instância do problema. O símbolo “=” pode ser usado para indicar que todos os veículos devem ser utilizados.

Os segundos e terceiros subcampos especificam as *características físicas* do veículo: a capacidade e a presença de compartimentos. A frota pode ser *homogênea* (todos os veículos têm a mesma capacidade) ou *heterogênea*. Há dois tipos de veículos compartimentados. Alguns veículos têm compartimentos intercambiáveis. Estes podem ser usados para separar mercadorias incompatíveis, tais como “galinhas e raposas”. Outros veículos possuem compartimentos especiais, usados para armazenar um tipo de mercadoria: por exemplo, comidas congeladas e vegetais frescos devem ser colocados em compartimentos especiais separados.

O quarto e o quinto subcampos especificam restrições temporais. É onde podem ser encontrados os intervalos de disponibilidade dos veículos e os limites inferiores e superiores na duração da rota.

<veículo>::=

<número de veículos>

<restrição de capacidade>

<restrição de mercadoria>

<restrição de disponibilidade do veículo>

<restrição de tempo de duração da rota>

<número de veículos> ::= < β_1 > < β_2 >

< β_1 > ::= $\circ \forall$ =

\circ [no máximo β_2 pode ser utilizado]

= [todos os β_2 veículos devem ser utilizados]

< β_2 > ::= $c \forall m$

c ($c \in \mathbb{N}$) [c veículos]

m [especificado como parte da instância do problema]

<restrição de capacidade> ::= $\circ \forall cap \forall cap_i$

\circ [nenhuma restrição de capacidade]

cap [veículos com capacidade idênticas]

cap_i [veículos com diferentes capacidades]

<restrição de mercadoria> ::= $\circ \forall sep \forall esp$

\circ [sem compartimento]

sep [veículos com compartimentos intercambiáveis]

esp [veículos com compartimentos especiais]

<restrição de disponibilidade do veículo> ::= $\circ \forall jt \forall jt_i$

\circ [nenhuma restrição de horário]

jt [limites de duração das rotas idênticos]

jt_i [limite de duração das rotas varia]

<restrição de tempo de duração das rotas> ::= $\circ \forall dur \forall dur_i$

\circ [não há restrições de tempo de duração das rotas]

dur [limites iguais de tempo de duração das rotas]

dur_i [diferentes limites de tempo de duração das rotas]

2.2.3 Características do Problema

O terceiro campo define o tipo de rede utilizada, a estratégia de serviço, e as restrições nas relações entre os endereços e os veículos.

O primeiro subcampo especifica as propriedades da rede (direcionada, não direcionada ou mista) e dos tempos de viagem (satisfazendo a métrica triangular ou não).

O segundo subcampo especifica a estratégia de serviço adotada. Há quatro tipos de estratégias de decisão:

- (1) O primeiro tipo permite ou não a quebra da demanda. A quebra da demanda ocorre quando é decidido, a princípio, que a demanda deve ser satisfeita por mais de uma visita ao cliente. Uma quebra posterior da demanda ocorre no caso de demanda estocástica onde, no momento em que o motorista chega ao local do cliente, ele descobre que a demanda é maior que a demanda prevista e decide não satisfazê-la completamente durante a visita.
- (2) No caso de roteirização em nós que envolve coletas e entregas, pode-se escolher por *backhauling*, isto é, as entregas são realizadas primeiro para esvaziar o veículo e depois realiza-se as coletas no caminho de volta para o depósito. No caso de roteirização tarefa, somente o carregamento de um cliente pode estar no veículo a qualquer hora.
- (3) Na maioria dos casos um veículo realiza no máximo uma rota por período, mas é possível permitir mais que uma rota por veículo.
- (4) Geralmente os veículos são restringidos a começar e terminar no mesmo depósito, mas isto pode ser relaxado de forma a permitir rotas multi-depósitos.

Os outros subcampos especificam possíveis relações entre dois endereços, entre um endereço e um veículo, ou entre dois veículos. Tais relações são causadas por um número de diferentes fatores, que enumerá-los aqui pode não ser

conveniente. Ao invés de descrevê-los, optou-se por especificar as restrições causadas por eles.

A mais conhecida destas relações é a *restrição de precedência* entre dois clientes: o veículo deve visitar um cliente antes de visitar o outro. Note que este requisito de precedência não tem nada a ver com a restrição implícita de precedência na roteirização de um par origem-destino, pois enxerga-se o par origem-destino como um único cliente.

A maioria das outras relações são restrições de *inclusão* e *exclusão*. Pode ser que um endereço deva ser visitado por um veículo partindo de determinado depósito, deva ser alocado em uma mesma rota que outro endereço específico, ou deva ser visitado por um veículo específico. Por exemplo, uma restrição de inclusão endereço-veículo ocorre se o veículo deve ser equipado com um dispositivo de descarregamento porque o cliente não possui uma doca para descarga da mercadoria. Pode ser também, que um endereço não possa ser atendido por um determinado depósito, ou não deva ser alocado na mesma rota que outro endereço, ou não deva ser visitado por um determinado veículo.

O último tipo de restrição é a *sincronização dos veículos*, que ocorre quando dois ou mais veículos devem trocar cargas ou assistir um ao outro.

<característica do problema>::=
 <tipo de rede>
 <tipo de estratégia>
 <restrição endereço-endereço>
 <restrição endereço-veículo>
 <restrição veículo-veículo>

<tipo de rede>::= $\langle \gamma_1 \rangle \langle \gamma_2 \rangle$
 $\langle \gamma_1 \rangle ::= \circ \vee \nabla$
 \circ [custo geral]

∇ [o custo satisfaz a métrica triangular]

$\langle \gamma_2 \rangle ::= \circ V \text{ dir } V \text{ mix}$

\circ [rede não direcionada]

dir [rede direcionada]

mix [rede direcionada e não direcionada]

$\langle \text{tipo de estratégia} \rangle ::= \langle \delta_1 \rangle \langle \delta_2 \rangle \langle \delta_3 \rangle \langle \delta_4 \rangle$

$\langle \delta_1 \rangle ::= \circ V / V \div$

\circ [não é permitida a quebra da demanda]

$/$ [é permitida a quebra da demanda no princípio]

\div [é permitida a quebra da demanda no final]

$\langle \delta_2 \rangle ::= \circ V \text{ back } V \text{ full}$

\circ [nenhuma restrição de *backhauls* ou carga completa]

back [*backhauls*, no caso de roteirização em nós]

compt [carga completa, no caso de roteirização tarefa]

$\langle \delta_3 \rangle ::= \circ V \geq 1R/v$

\circ [no máximo uma rota por veículo]

$\geq 1R/v$ [permite mais que uma rota por veículo]

$\langle \delta_4 \rangle ::= \circ V \geq 1D/r$

\circ [uma rota começa e termina no mesmo depósito]

$\geq 1D/r$ [permite rotas multi-depósitos]

$\langle \text{restrição endereço-endereço} \rangle ::= \langle \epsilon_1 \rangle \langle \epsilon_2 \rangle \langle \epsilon_3 \rangle$

$\langle \epsilon_1 \rangle ::= \circ V \text{ prec}$

\circ [sem restrição de precedência]

prec [com restrição de precedência]

$\langle \epsilon_2 \rangle ::= \circ V \text{ DE}$

\circ [sem restrição depósito-endereço]

DE [com restrição depósito-endereço]

$\langle \epsilon_3 \rangle ::= \circ V \text{ EE}$

\circ [sem restrição endereço-endereço]

EE [com restrição endereço-endereço]

$\langle \text{restrição endereço-veículo} \rangle ::= \langle \xi_1 \rangle \langle \xi_2 \rangle$

$\langle \xi_1 \rangle ::= \circ \vee DV$

\circ [sem restrição depósito-veículo]

DV [com restrição depósito-veículo]

$\langle \xi_2 \rangle ::= \circ \vee EV$

\circ [sem restrição endereço-veículo]

EV [com restrição endereço-veículo]

$\langle \text{restrição veículo-veículo} \rangle ::= \circ \vee VV$

\circ [sem restrição veículo-veículo]

VV [sincronização necessária entre veículos]

2.2.4 Objetivos

O quarto campo define as funções objetivo. Para especificar uma função objetivo são introduzidos cinco elementos quantitativos.

O tempo de viagem e de atendimento do veículo i , isto é, a duração da rota deste veículo será denotada por T_i . Com este quantitativo se expressa o objetivo padrão encontrado na literatura de roteirização e programação de veículos: minimizar o tempo total de viagem e atendimento.

Para ser capaz de expressar funções objetivo mais realistas, é introduzida uma função custo do veículo C_i , uma função custo no endereço C_j , uma função penalidade do veículo P_i , e uma função penalidade do endereço P_j . Uma função custo do veículo pode ser usada para modelar situações onde, adicionada à roteirização e programação, é requisitada também a determinação do tamanho e tipo de frota. Uma função custo no endereço leva em consideração o custo incorrido ao divergir do nível de serviço pretendido. As funções penalidades permitem modelar os custos incorridos com a violação de restrições flexíveis. Há restrições que podem ser violadas até um certo custo; horas extras de motoristas são um exemplo. As

restrições que são consideradas flexíveis são listadas como argumentos das funções penalidades do veículo e endereço.

Na prática, os problemas geralmente têm uma função objetivo composta. O usuário pode especificá-la pela listagem dos componentes da função objetivo em ordem decrescente de importância. Para outros fins, este esquema também deixa em aberto a possibilidade de não especificar nenhum objetivo, reduzindo o problema para uma questão de ordem prática.

Nas regras seguintes, nota-se que, no caso de um único veículo, os operadores *sum* ou *max* e o subscrito *i* são retirados dos objetivos relacionados à rota e ao veículo.

<objetivo> ::= <operador> <função>

<operador> ::= sum V max

sum [minimizar a soma do valor da função custo]

max [minimizar o valor da função custo máximo]

<função> ::= $T_i \vee C_i \vee P_i$ (<restrições do veículo>) $\vee C_j \vee P_j$ (<restrições no endereço>)

T_i [duração da rota]

C_i [custo do veículo]

P_i (<restrições do veículo>) [penalidade no veículo]

C_j [custo no endereço]

P_j (<restrições no endereço>) [penalidade no endereço]

<restrição do veículo> ::= $cap \vee cap_i \vee jt \vee jt_i \vee dur \vee dur_i$

<restrições no endereço> ::= $jt_j \vee jm_j$

A intenção deste capítulo foi mostrar que, primeiramente, o problema de roteirização de veículos possui muitas variações. Em segundo lugar, existem na literatura várias propostas para classificar as variações deste problema, onde cada qual segue uma orientação ou estrutura diferente. Por uma questão de padronização

ou maior formalização no desenvolvimento do trabalho, a estrutura para classificação do problema proposta por Desrochers *et al.* (1990) foi adotada para classificar ou especificar tais variações, quando necessário. Antes de partir para o desenvolvimento dos objetivos deste estudo é importante tratar dos aspectos computacionais envolvidos com estes problemas, tema do capítulo seguinte.

3 COMPLEXIDADE COMPUTACIONAL, ESTRATÉGIAS E MÉTODOS DE SOLUÇÕES

3.1 COMPLEXIDADE COMPUTACIONAL

O desenvolvimento do conceito de *complexidade computacional* é uma importante consideração na formulação e resolução de problemas de roteirização e programação de veículos. Estes problemas, e muitos outros como, por exemplo, o problema do caixeiro viajante, pertencem a uma classe maior conhecida como problemas combinatoriais. Os problemas de natureza combinatorial são estruturados em função de uma variável de decisão, sujeita à algumas limitações ou restrições. Generalizando, eles podem ser formulados da seguinte forma:

$$\begin{array}{ll} \text{Minimizar} & f(x) \\ \text{Sujeito à:} & g_i(x) \leq b_i; \quad i=1, \dots, m; \\ & h_j(x) = c_j; \quad j=1, \dots, n. \end{array}$$

Onde, x é um vetor de variável de decisão discreta (isto é, a solução é um conjunto ou uma seqüência de inteiros ou outros objetos discretos) e $f(\cdot)$, $g_i(\cdot)$ e $h_j(\cdot)$ são funções gerais. Esta formulação é dada para um problema de minimização, mas as modificações necessárias para transformá-la em um problema de maximização podem ser feitas.

É importante deixar claro que o uso do termo *problema* é genérico. Em uma situação real, onde se tem um problema particular, os símbolos usados para descrevê-

lo têm valores numéricos específicos. Costuma-se usar o termo *instância* para distinguir uma situação particular de uma geral. Assim, um *problema* será uma questão geral a ser respondida, descrito por: (1) uma declaração geral de todos os seus parâmetros; e (2) uma declaração de quais propriedades a resposta (ou solução) deve satisfazer. Uma *instância* do problema é obtida especificando-se valores particulares para todos os parâmetros do problema.

Como um exemplo, considere o clássico “problema do caixeiro viajante - PCV” (Lenstra e Rinnooy Kan, 1981). Os parâmetros deste problema consistem em um conjunto finito de $C = \{c_1, c_2, \dots, c_m\}$ de “cidades”, e para cada par de cidades c_i, c_j em C , a “distância” $d(c_i, c_j)$ entre elas. Uma solução é uma seqüência $\langle c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(m)} \rangle$ das cidades dadas, que minimize:

$$[\sum d(c_{\pi(i)}, c_{\pi(i+1)})] + d(c_{\pi(m)}, c_{\pi(1)})$$

onde, $c_{\pi(i)}$ = cidade que está na posição i da seqüência.

Esta expressão dá o comprimento de uma rota que começa em $c_{\pi(1)}$ (depósito), visita cada cidade na seqüência e, da última cidade $c_{\pi(m)}$ retorna para $c_{\pi(1)}$. Uma instância deste problema (ilustrada na figura 1) é dada por $C = \{c_1, c_2, c_3, c_4\}$, $d(c_1, c_2) = 10$, $d(c_1, c_3) = 5$, $d(c_1, c_4) = 9$, $d(c_2, c_3) = 6$, $d(c_2, c_4) = 9$, e $d(c_3, c_4) = 3$.

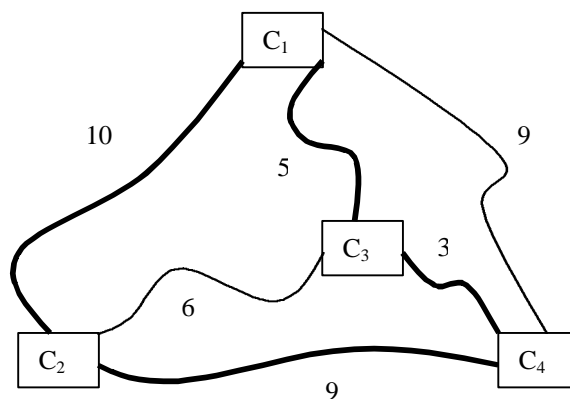


FIGURA 1- UMA INSTÂNCIA DO PROBLEMA DO CAIXEIRO VIAJANTE E A ROTA MÍNIMA PARA ESTE CASO.

Fonte: Garey e Johnson, 1979

A seqüência $\langle c_1, c_2, c_4, c_3 \rangle$ é uma solução para esta instância, e sua rota corresponde a rota mínima possível de comprimento 27.

Uma maneira de resolver um problema combinatorial consiste simplesmente em listar todas as suas soluções possíveis, avaliar sua função objetivo, e escolher a melhor solução. Em princípio, este método de *enumeração completa* permite resolver qualquer problema, mas na prática isto se torna impossível devido ao grande número de soluções existentes para qualquer problema de tamanho razoável. Para ilustrar este fato, considere o problema do caixeiro viajante com N cidades. Como o ponto de partida é arbitrário, existem $(N - 1)!$ soluções possíveis (PCV assimétrico) ou $(N - 1)!/2$ se as distâncias entre os pares de cidades são as mesmas, independentemente do sentido viajado (PCV simétrico). Suponha que um computador possa listar todas as possíveis soluções de um problema de 20 cidades em uma hora. Utilizando a fórmula acima, o mesmo computador levaria 20 horas para listar todas as soluções de um problema de 21 cidades; 17,5 dias para resolver um problema de 22 cidades; e um problema de 25 cidades poderia levar aproximadamente 6 séculos.

Vários algoritmos exatos foram inventados para achar a solução ótima de um problema de maneira mais eficiente do que enumeração completa.

Algoritmos são geralmente procedimentos para resolver problemas “passo-a-passo”. O significado de algoritmo “mais eficiente” esta normalmente ligado a sua rapidez de solução, ou ainda, o tempo necessário para um algoritmo resolver um dado problema é o fator dominante que determina se ele é eficiente o bastante para ser utilizado na prática.

Este tempo é convenientemente expresso em termos de uma única variável, o “tamanho” de uma instância do problema, o que reflete a quantidade de dados de entrada necessários para descrever a instância. Frequentemente o tamanho de uma instância é medido de uma maneira informal. Para o problema do caixeiro viajante, por exemplo, é comum usar o número de cidades. No entanto, uma instância do

problema de m cidades (PCV simétrico) inclui, além da identificação das m cidades, uma quantidade de $m(m-1)/2$ números que definem as distâncias entre as cidades, e a magnitude destes números também contribui com a quantidade de dados. Para expressar o tempo de maneira precisa, devem ser levados em consideração todos estes fatores na definição do tamanho da instância.

Diferentes algoritmos possuem uma grande variedade de diferentes funções que definem sua complexidade de tempo computacional. A *função de complexidade de tempo* de um algoritmo expressa o seu pior desempenho possível, ou seja, o tempo máximo necessário para que ele resolva a “pior” instância do problema.

Um *algoritmo de tempo polinomial* é definido por uma função de complexidade de tempo $O(p(n))$, onde O é uma constante, p é uma função polinomial e n é usado para denotar o tamanho da instância do problema. Qualquer algoritmo cuja função de complexidade de tempo não possa ser assim limitada é chamado *algoritmo de tempo exponencial* (esta definição também inclui certas funções de complexidade de tempo não polinomiais, como $n^{\log n}$, que normalmente não é considerada uma função exponencial).

A distinção entre estes dois tipos de algoritmos tem um significado particular quando é considerada a solução de grandes instâncias do problema. A tabela 1 ilustra as diferenças nas taxas de crescimento no tempo, entre os tipos de funções complexidade típicas, onde a função expressa o tempo de execução em microsegundos. Note que a taxa de crescimento para a função de complexidade exponencial é muito mais explosiva.

Esta tabela indica uma das razões pelas quais um algoritmo de tempo polinomial é mais desejável que um algoritmo de tempo exponencial.

Tabela 1: Comparação entre várias funções de complexidade de tempo polinomial e exponencial.

FUNÇÃO DE COMPLEXIDADE DE TEMPO	TAMANHO					
	n					
	10	20	30	40	50	60
n	0,00001	0,00002	0,00003	0,00004	0,00005	0,00006
	segundos	segundos	segundos	segundos	segundos	segundos
n ²	0,0001	0,0004	0,0009	0,0016	0,0025	0,0036
	segundos	segundos	segundos	segundos	segundos	segundos
n ³	0,001	0,008	0,027	0,064	0,125	0,216
	segundos	segundos	segundos	segundos	segundos	segundos
n ⁵	0,1	3,2	24,3	1,7	5,2	13,0
	segundos	segundos	segundos	minutos	minutos	minutos
2 ⁿ	0,001	1,0	17,9	12,7	35,7	366
	segundos	segundos	minutos	dias	anos	séculos
3 ⁿ	0,059	58	6,5	3855	2 x 10 ⁸	1,3 x 10 ¹³
	segundos	minutos	anos	séculos	séculos	séculos

Fonte: Garey e Johnson, 1979.

De acordo com Garey e Johnson (1979), a natureza fundamental desta distinção foi primeiramente discutida em Cobham (1964⁶) e Edmonds (1965⁷). Edmonds, em particular, comparou algoritmos de tempo polinomial com “bons” algoritmos e presumiu que certos problemas de programação inteira não podem ser resolvidos pelos tais “bons” algoritmos. Isto reflete o ponto de vista de que algoritmos de tempo exponencial não devem ser considerados “bons” algoritmos, o que de fato geralmente acontece. A maioria dos algoritmos de tempo exponencial são meras variações de busca exaustiva e sua performance não se mostra melhor do que enumeração completa. Por outro lado, algoritmos de tempo polinomial são geralmente possíveis de serem criados através da percepção de ganho na estrutura de um problema. Assim, um problema é considerado *intratável* quando ele for tão difícil, que não se conhece nenhum algoritmo de tempo polinomial para resolvê-lo.

A distinção entre “eficientes” algoritmos de tempo polinomial e “ineficientes” algoritmos de tempo exponencial admite muitas exceções quando a instância do problema de interesse tem tamanho limitado. Na tabela 1, por exemplo, o algoritmo

⁶ COBHAM, A. (1964). “The intrinsic computational difficulty of functions”. In: INTERNATIONAL CONGRESS FOR LOGIC METHODOLOGY AND PHILOSOPHY OF SCIENCE, Bar-Hillel, Y. (ed.). North Holland, Amsterdam, Proc. 1964, p. 24-30.

⁷ EDMONDS, J. (1965). “Paths, trees, and flowers”. Canadian Journal of Mathematics, v.17, p. 449-467.

2^n é mais rápido que o algoritmo n^5 para $n \leq 20$. Outros exemplos podem ser facilmente construídos.

Além do mais, existem alguns algoritmos de tempo exponencial que têm sido muito úteis na prática. Complexidade de tempo como definição, é uma medida de *pior-caso*, e o fato de que um algoritmo tem complexidade de tempo 2^n significa que pelo menos uma instância do problema de tamanho n requer tanto tempo.

A maioria das instâncias do problema pode, na realidade, necessitar de menos tempo. O algoritmo *simplex* para programação linear, por exemplo, tem mostrado ter complexidade exponencial, mas possui um tempo de processamento muito rápido na prática. Da mesma forma, o algoritmo *branch-and-bound* para o problema da mochila, que também possui complexidade de tempo exponencial, tem tido tanto sucesso que muitos especialistas o consideram um problema “bem resolvido”.

Infelizmente, exemplos como estes são muito raros. Algoritmos de tempo exponencial são conhecidos para muitos problemas, mas poucos deles são úteis na prática, o que leva à contínua busca por algoritmos de tempo polinomial. O grande sucesso deste tipo de algoritmo é conduzido pela suspeita de que eles capturam uma propriedade crucial dos problemas, cujo refinamento pode levar, de alguma maneira, a melhores métodos. Pouco progresso foi feito no sentido de explicar esta suspeita, e nenhum método é conhecido para dizer com antecedência o quão rápido um dado algoritmo de tempo exponencial irá rodar na prática.

Por outro lado, o tempo de execução limitado por algoritmos de tempo polinomial geralmente permite tal previsão. Os problemas de natureza polinomial geralmente possuem tempo de execução limitado por polinômios de grau 2 ou 3, no pior dos casos, e não envolvem grandes coeficientes. Algoritmos que satisfazem tais limites podem ser considerados “provavelmente eficientes”, e esta é a propriedade mais desejada e que faz dos algoritmos de tempo polinomial o método preferido para resolver os problemas.

3.1.1 P e NP

Da mesma maneira que um algoritmo é classificado como bom ou ruim, dependendo se ele possui ou não complexidade de tempo polinomial, é possível classificar um *problema* como “difícil” ou “fácil” dependendo se ele pode ou não ser resolvido por um algoritmo de complexidade de tempo polinomial.

Por uma questão de uniformidade, a teoria da complexidade computacional se restringe à *problemas de decisão*, isto é, problemas cujas soluções requerem apenas um “sim” ou “não” como resposta. É importante deixar claro que isto não restringe a aplicabilidade da teoria. Problemas de otimização são naturalmente expressos como problemas de decisão e, se existe um algoritmo de tempo polinomial para o problema de decisão, então também o existe para a sua versão em problema de otimização.

Por exemplo, um problema de decisão relacionado ao problema do caixeiro viajante pode ser descrito da seguinte forma:

INSTÂNCIA: um conjunto finito $C = \{c_1, c_2, \dots, c_m\}$ de “cidades”, uma “distância” $d(c_i, c_j) \in \mathbb{Z}^+$ para cada par de cidades $c_i, c_j \in C$, e um limitante $B \in \mathbb{Z}^+$ (onde \mathbb{Z}^+ denota os inteiros positivos).

$$[\sum d(c_{\pi(i)}, c_{\pi(i+1)})] + d(c_{\pi(m)}, c_{\pi(1)}) \leq B ?$$

QUESTÃO: existe uma “rota” com todas as cidades de C , que é uma seqüência $\langle c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(m)} \rangle$, tal que possui distância total não maior que B ?

Este exemplo serve para ilustrar como um problema de decisão pode derivar de um problema de minimização, incluindo um limite numérico B como um parâmetro adicional. Problemas de decisão também podem derivar de problemas de maximização, de forma análoga, substituindo “não mais que” por “pelo menos”. Mesmo que a teoria tenha sido desenvolvida apenas para os problemas de decisão,

pode-se estender suas implicações sobre os problemas de otimização. De acordo com o teorema 1 em Johnson e Papadimitriou (1985), existe um algoritmo de tempo polinomial para um problema de otimização, se e somente se, existe um algoritmo de tempo polinomial para a sua versão em problema de decisão.

Tecnicamente, são ditos pertencentes à classe P (*Polynomial time*), todos os problemas de decisão para os quais são conhecidos algoritmos de tempo polinomial.

A classe NP (*Non-deterministic Polynomial time*) consiste de problemas que são tidos como “difíceis”, e portanto só podem ser resolvidos em tempo polinomial, através de algoritmos não determinísticos.

O conceito de algoritmo polinomial não determinístico é bastante sutil, mas pode ser definido intuitivamente da seguinte forma. Primeiro, deve-se ter em mente que os problemas em NP são problemas de decisão. Agora, imagine um computador que tem a propriedade de, a cada vez que uma escolha deva ser feita ou uma decisão deva ser tomada, se dividir em duas “cópias”, sendo cada “cópia” explorada em paralelo. Esta propriedade é bastante parecida com uma “busca em árvore”, onde todas as ramificações podem ser exploradas simultaneamente. Se (e somente se) uma das cópias trouxer uma resposta afirmativa, o problema de decisão terá sido resolvido. Se a busca em cada ramificação levar um tempo máximo limitado por um polinômio, então o problema está na classe NP.

Uma outra maneira de ver isto é supor que se possa “adivinhar” uma solução para o problema, e é requerido que a sua *verificação* seja levada a cabo em tempo polinomial. Isto é possível devido ao potencial que os algoritmos não determinísticos possuem em computar um número exponencial de cálculos em paralelo, após um número polinomial não determinístico de “passos”.

O conceito de *transformabilidade* (algumas vezes chamado *reduzibilidade*) também é importante. Suponha que se tenha um problema Π_1 que possa ser resolvido por um algoritmo A. Se for possível transformar qualquer instância de um outro

problema Π_2 em uma instância de Π_1 em tempo polinomial, então pode-se utilizar este fato, mais o algoritmo A, para resolver Π_2 . A classe de instâncias de Π_1 é, pelo menos tão grande quanto (e provavelmente maior) que a classe das instâncias de Π_2 – estas últimas são de certo modo “casos especiais” entre as instâncias de Π_1 . Então é razoável considerar Π_1 como sendo pelo menos tão difícil quanto (e provavelmente mais difícil que) Π_2 .

Se todos os problemas pertencentes à classe NP puderem ser transformados em um problema Π , então Π é um problema *NP-hard*. Adicionalmente, se o problema Π pertence à NP, Π é dito *NP-completo*.

Os problemas *NP-completo* têm sido considerados foco de interesse prático e teórico nos últimos anos. Esta classe de problemas contém muitos problemas conhecidos, como por exemplo, o problema do caixeiro viajante e muitos outros, alguns dos quais estão listados em Garey e Johnson (1979), e é caracterizada por duas importantes propriedades:

- 1) Não é conhecido nenhum algoritmo de tempo polinomial que possa resolver um problema *NP-completo*.
- 2) Se houver algum algoritmo de tempo polinomial para um dos problemas *NP-completo*, pode-se obter algoritmos de tempo polinomial para todos os problemas *NP-completo*.

Esta implicação da teoria do *NP-completo* pode ser estendida além dos problemas de decisão, e portanto além dos NP, através do uso mais geral de transformação entre problemas.

Algumas vezes encontra-se na literatura uma confusão de terminologias. Um problema de *otimização* não está dentro de NP, mesmo que a sua versão em problema de decisão esteja. Embora a precisão na terminologia utilizada seja importante, quando o leitor se deparar com uma frase do tipo “este problema (de otimização) é *NP-completo*”, o autor está provavelmente assumindo que o leitor

entenda que “o problema de decisão correspondente a este problema de otimização é *NP-completo*”. Todavia, muitos autores fazem uso do termo *NP-hard* para descrever tais problemas de otimização, e esta prática é preferida.

Uma relação que não foi declarada explicitamente, mas que pode ser concluída é que $P \subseteq NP$. Todo problema de decisão que é resolvido por algoritmos de tempo polinomial também pode ser resolvido por algoritmos não determinísticos de tempo polinomial. Um algoritmo de tempo polinomial A que resolve o problema $\Pi \in P$, pode ser utilizado no estágio de *verificação* de um algoritmo não determinístico de tempo polinomial para resolver Π . Assim, $\Pi \in P$ implica que $\Pi \in NP$. No entanto, existem muitos problemas em NP para os quais não é conhecido nenhum algoritmo que os resolvam em tempo polinomial.

Por isso, acredita-se que $P \neq NP$, mesmo não havendo provas para isto. Todas as tentativas teóricas para provar que $P=NP$ falharam porque nenhum algoritmo exato de tempo polinomial foi encontrado para nenhum problema em NP . Sendo assim, é mais razoável (dado o estágio atual de conhecimento) operar sobre a consideração de que $P \neq NP$, embora continue existindo a possibilidade, mesmo que pequena, de alguém provar que $P = NP$.

A figura 2 contém uma tentativa de representação do mundo NP .

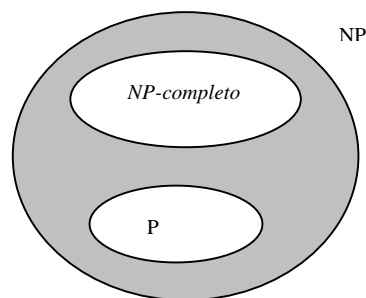


FIGURA 2 - TENTATIVA DE REPRESENTAÇÃO DO MUNDO NP

Fonte: Garey e Johnson, 1979.

Note que NP não é simplesmente dividido em *NP-completo* e P. Se P difere de NP, então devem existir problemas em NP que não são *NP-completo* e nem resolvidos em tempo polinomial. No entanto, o interesse principal corresponde aos problemas *NP-completo*.

Segundo Lenstra e Rinnooy Kan (1981), quase todos os problemas de roteirização e programação de veículos, assim como a maioria dos problemas de otimização combinatorial, são *NP-hard*. A teoria da complexidade computacional tem mostrado forte evidência de que qualquer algoritmo exato de otimização desta natureza terá, provavelmente, uma performance ruim quanto ao tempo de processamento, em algumas ocasiões. Mais formalmente, no pior caso, o tempo de processamento irá crescer exponencialmente com o tamanho do problema. Este fato vem resultando no desenvolvimento de uma variedade de algoritmos de aproximação ou heurísticas para a solução de problemas de roteirização e programação de veículos. Um algoritmo heurístico é um procedimento que emprega a estrutura do problema em um caminho matemático (geralmente intuitivo) para obter uma solução factível ou aproximadamente ótima.

3.2 ESTRATÉGIAS E MÉTODOS DE SOLUÇÕES

Bodin *et al.* (1983) classificaram as estratégias de solução para os problemas de roteirização de veículos da seguinte forma:

- 1) *Agrupar – Roteiriza (cluster first - route second)*: consiste no procedimento de agrupar os nós ou arcos de demanda primeiro, e depois construir rotas econômicas para cada agrupamento. Exemplos destas idéias são aplicadas nos trabalhos de Gillett e Miller (1974¹), Gillett e Johnson (1976²), Chapleau *et al.*

¹ GILLET, B.; MILLER, L. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operational Research*, n.22, p. 340-349.

² GILLET, A.; JOHNSON, J. (1976). Multi-terminal vehicle dispatch algorithm. *Omega*, n.4, p. 711-718

(1981³) e Karp (1977⁴) para o problema básico de roteirização de veículos com um único depósito.

- 2) *Roteiriza – Agrupa (route first – cluster second)*: primeiro, uma grande (geralmente inactível) rota ou ciclo é construída incluindo todas as entidades de demanda (nós e/ou arcos). Depois esta grande rota é dividida em um número menor e factível de rotas. Golden *et al.* (1982⁵) desenvolveram um algoritmo que utiliza este conceito para o problema de roteirização com frota heterogênea de veículos. Newton e Thomas (1969⁶) e Bodin e Berman (1979⁷) utilizaram esta estratégia para roteirização de ônibus escolares de e para uma única escola. Bodin e Kursh (1978⁸ e 1979⁹), utilizaram esta estratégia para o problema de varrição de ruas.
- 3) *Economias ou Inserções*: procedimento de construção de uma solução em um dado caminho de forma que, para cada etapa do processo, a configuração em construção, possivelmente inactível, é comparada com uma configuração alternativa que deverá ser inactível também. A configuração alternativa é aquela que apresentar a maior economia em termos de alguma função ou critério adotado, tal como custo total, ou a que inserir de forma menos custosa entidades de demanda ainda não inseridas na rota (ou rotas) em construção existente. O processo é finalmente concluído com uma configuração factível. Exemplos de

³ CHAPLEAU, L.; FERLAND, J.; ROUSSEAU, J. M. (1981). *Clustering for routing in dense area*. University of Montreal. Transportation Research Center Publication, n. 206.

⁴ KARP, R. (1977). Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. *Mathematics of Operations Research*, n.2, p. 209-224.

⁵ GOLDEN, B.; ASSAD, A.; LEVY, L.; GHEYSENS, F. (1982). The fleet size and mix vehicle routing problem. *Management Science & Statistic Working Paper*, n.82-020.

⁶ NEWTON, R.; THOMAS, W. (1969). Design of school bus routes by computer. *Socio-Economic Planning Science*, n.3, p.75-85.

⁷ BODIN, L.; BERMAN, L. (1979). Routing and scheduling of school buses by computer. *Transportation Science*, n.13, p. 113-129.

⁸ BODIN, L.; KURSH, S. (1978). A computer-assisted system for the routing and scheduling of street sweepers. *Operational Research*, v. 26, n.4, p. 525-537

⁹ BODIN, L.; KURSH, S. (1979). A detailed description of a street sweeper routing and scheduling system. *Computers & Operations Research*, n.14B, p. 115-120.

procedimentos de economia/inserção são descritas por Clarke e Wright (1964¹⁰), Golden *et al.* (1977¹¹), Norback e Love (1979¹²), e Golden e Wong (1981¹³). Hinson e Mulherkar (1975¹⁴) usaram uma variação deste procedimento para roteirização de aviões.

- 4) *Melhoria/Troca*: procedimento heurístico também conhecido como troca de arcos ou arestas onde em cada etapa uma solução factível é alterada, resultando em outra solução factível com o custo total reduzido. Este processo continua até que não seja mais possível reduções adicionais no custo. Procedimentos de troca são também conhecidos como procedimentos *r-opt*, onde *r* é o número de arcos ou arestas trocados a cada iteração. Em um algoritmo *r-opt*, todos os *r* arcos são testados até não existir mais nenhuma troca factível que melhore a solução. O número de operações necessárias para testar todas as *r* trocas cresce rapidamente com o aumento do número de cidades. Assim, valores de *r* = 2 (*2-opt*) e *r* = 3 (*3-opt*) são os mais usados. Bodin e Sexton (1979¹⁵), modificaram este método para programação de microônibus no problema de *dial-a-ride* (programa de transporte público pré-agendado por telefone).
- 5) *Programação matemática*: inclui algoritmos que são diretamente baseados em uma formulação de programação matemática de problemas de roteirização subordinados. Uma discussão mais profunda deste assunto pode ser encontrado em Magnanti (1981¹⁶).

¹⁰ CLARKE, G.; WRIGHT, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operational Research*, n.12, p. 568-581.

¹¹ GOLDEN, B.; MAGNANTI, T.; NGUYEN, H. (1977). Implementing vehicle routing algorithms. *Networks*, n.7, p. 113-148.

¹² NORBACK, J.; LOVE, R. (1979). Geometric approaches to solve the traveling salesman problem. *Management Science*, n.23, p. 1208-1223.

¹³ GOLDEN, B.; WONG, R. (1981). Capacitated arc routing problems. *Networks*, n. 11(3), p. 305-315.

¹⁴ HINSON, J.; MULHERKAR, S. (1975). *Improvements to the Clarke and Wright algorithm as applied to an airline scheduling problem*. Technical Report, Federal Express Corporation.

¹⁵ BODIN, L.; SEXTON, T. (1979). *The subscriber dial-a-ride problem*. Maryland, College Park, University of Maryland, College of Business and Management, (Report n.UMCP-UMTA-1-79).

¹⁶ MAGNANTI, T. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, v.11, n.2, p. 179-214.

- 6) *Otimização interativa*: procedimento de propósito geral no qual há um alto grau de interação humana incorporado no processo de solução do problema. A idéia é que o tomador de decisões tem a experiência e a capacidade de colocar os parâmetros revisados e injetar correções subjetivas baseadas no conhecimento e na intuição do modelo de otimização. Isto quase sempre aumenta a probabilidade de que o modelo será implementado e usado. Algumas adaptações iniciais deste método para problemas de roteirização de veículos são apresentadas por Krolak *et al.* (1971¹⁷). Cullen *et al.* (1981¹⁸) introduziram novas heurísticas de otimização interativa.
- 7) *Procedimento exatos*: Com exceção do problema do caixeiro viajante, os esforços no desenvolvimento de algoritmos exatos para problemas de roteirização e programação do tipo *NP-Hard* têm sido muito limitados. Estas exceções incluem o método de particionamento para programação de pessoal e algoritmos exatos para programação de veículos (desenvolvido por Christofides *et al.*, 1981) utilizando técnicas especializadas de *branch and bound* e programação dinâmica. Técnicas de soluções exatas para problemas de roteirização e programação de veículos continuam em um estágio muito inicial do seu desenvolvimento.

Laporte (1992), Christofides (1985) e Osman (1993) classificaram os métodos de soluções em algoritmos exatos e heurísticas, enquanto Cunha (1997) propôs a classificação dos métodos de solução nas seguintes categorias, que serão discutidas na seqüência deste capítulo:

- Métodos exatos, que garantem a obtenção da solução ótima.
- Métodos heurísticos, que não garantem a solução ótima, mas geralmente resultam em soluções sub-ótimas de grande qualidade a um esforço computacional menor.

¹⁷ KROLAK, P.; FELTS, W.; MARBLE, G. (1972). A man-machine approach toward solving the traveling salesman problem. *Transportation Science*, n.6, p. 146-170.

¹⁸ CULLIEN, F.; JARVIS, J.; RATLIFF, H. (1981). Set partitioning based heuristics for interactive routing. *Network*, n.11, v.2, p. 125-144.

- Métodos emergentes, que reúnem as técnicas mais recentes e avançadas, não tradicionais, baseadas em sistemas especialistas, ou em métodos de busca ou iterativos (por exemplo, metaheurísticas, algoritmos genéticos, busca tabu, etc.).

Segundo Souza (1993¹⁹) *apud* Cunha (1997), embora estes últimos possam ser considerados métodos heurísticos, na medida em que não se assegura a obtenção da solução ótima, foram considerados separadamente, uma vez que se caracterizam por utilizar alternadamente dois ou mais procedimentos em uma ordem definida, a fim de modificar ou aprimorar gradualmente uma solução.

3.2.1 Algoritmos Exatos

Como uma generalização do problema do caixeiro viajante, o problema de roteirização de veículos pertence à classe dos *NP-hard*, e algoritmos de tempo polinomial para achar a solução ótima são improváveis de existir. Por isso, pouca atenção tem sido dada à busca de soluções ótimas (Osman, 1993).

Christofides (1985) apresenta três formulações que tem sido utilizadas como base para os métodos de soluções dos PRV. Duas destas formulações consistem em programação inteira, enquanto a terceira consiste em formulação para programação dinâmica. Fisher e Jaikumar (1978) utilizaram a formulação para programação inteira do PRV básico na construção de um algoritmo de otimização baseado na decomposição de Bender, além de utilizar técnicas de *branch and bound* e relaxação lagrangiana.

Christofides *et. al* (1980 e 1981) definiram o problema de roteirização de veículos, adaptando o problema de particionamento de conjunto (*set covering* ou *partitionig problems*) com uma restrição a mais, através de uma formulação para programação inteira. Em uma outra tentativa, os autores utilizaram a formulação de programação dinâmica do PRV básico e técnicas de relaxação lagrangiana (*state-*

¹⁹ SOUZA, P. S. (1993). *Asynchronous organizations for multi-algorithms problems*. Pittsburgh. Tese (Doutorado) - Carnegie Mellow University, Department of Electrical and Computer Engineering.

space relaxation) para calcular o limite usado no algoritmo de *branch and bound* para o PRV.

3.2.2 Métodos de aproximação ou heurísticas

Reeves (1993) define *heurística* como uma técnica que busca boas (isto é, perto da ótima) soluções, com um custo computacional razoável, sem garantir soluções factíveis ou ótimas, e em muitos casos não é capaz de declarar o quão próximo uma solução factível está da solução ótima.

Além da teoria da complexidade computacional representar uma forte justificativa para a utilização de métodos heurísticos na solução de PRV, outro forte argumento apresentado pelo autor corresponde a possibilidade de modelar o problema real com maior precisão, uma vez que as heurísticas são mais flexíveis e aptas a operar com funções objetivos e/ou restrições mais complicadas (e mais realistas) do que os algoritmos exatos. Colocando isto de uma outra forma, o que traz mais benefícios: uma solução exata de um modelo aproximado ou uma solução aproximada de um modelo exato?

Christofides (1985) classifica as heurísticas para roteirização de veículos nas seguintes categorias:

- Métodos construtivos;
- Método das duas-fases;
- Método de otimização incompleta.

Os *métodos construtivos* ainda podem se diferenciar de acordo com a maneira como a rota é construída (seqüencial ou paralela) e o critério utilizado para expandir a rota.

O método construtivo mais utilizado consiste no método de economia (ou inserção) de Clarke e Wright (1964). Este método inicia com uma solução inviável

em que cada cliente é atendido individualmente por um veículo e procede da seguinte forma:

Passo 1: calcula-se a economia s_{ij} que pode ocorrer no custo se o *link* (i, j) for construído, para todos os pares de clientes i e j (ver figura 3), resultando em uma nova rota $(1, i, j, 1)$ ao invés de atender i e j em duas rotas $(1, i, 1)$ e $(1, j, 1)$. Note que $s_{ij} = c_{i1} - c_{ij} + c_{1j}$.

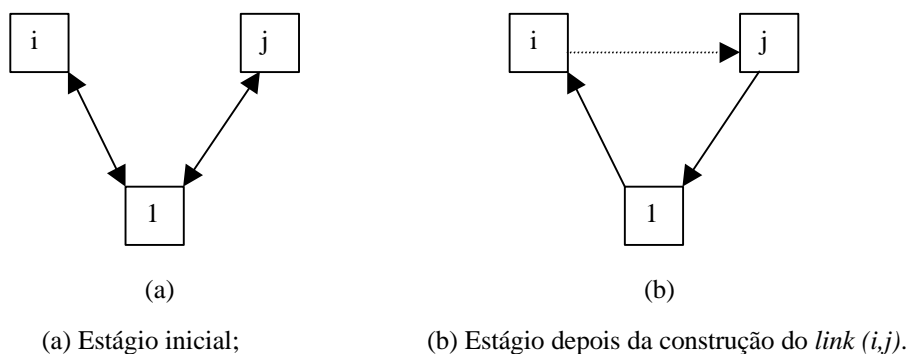


FIGURA 3- INSERÇÃO DO *LINK* (i, j) .

Fonte: Christofides, 1985.

Passo 2: listar as economias em ordem decrescente.

Passo 3: iniciando do topo da lista, fazer:

Versão Paralela

Passo 4: se ao incluir o *link* na rota, resultar em uma rota factível de acordo com as restrições do PRV, então anexar este *link* à solução; se não, rejeitar o *link*.

Passo 5: tentar com o próximo *link* da lista e repetir o passo 4 até que todos os *links* já tenham sido investigados.

Versão Seqüencial

Passo 4: buscar o primeiro *link* factível da lista, que pode ser usado para estender um dos dois extremos da rota em construção.

Passo 5: se a rota não pode ser mais estendida, terminar a rota. Escolher o primeiro *link* factível na lista para começar uma nova rota.

Passo 6: repetir os passos 4 e 5 até que nenhum *link* possa ser escolhido.

Muitas modificações tem sido propostas para o método de Clarke e Wright (1964), buscando diferentes resultados. Gaskell (1967) e Yellow (1970) introduziram um novo conceito de economia dado por $s_{ij} - \mathbf{q} c_{ij}$, onde \mathbf{q} é um parâmetro escalar. Variando \mathbf{q} , é possível dar maior ou menor ênfase aos custos de viagem entre dois nós, dependendo da sua posição relativa ao depósito. Este parâmetro pode ser alterado, obtendo-se diferentes soluções.

Golden *et al.* (1977) reduziram substancialmente o tempo de processamento da heurística de Clarke e Wright, utilizando técnicas de ciências computacionais. Outras heurísticas de construção podem ser encontradas em Mole e Jameson (1976), Nelson *et al.* (1985), Paessens (1988), Altikemer e Gavish (1991).

Desrochers e Verhoog (1990) apresentam uma nova heurística de construção para o PRV com frota homogênea, baseada em sucessivas fusões de rotas. A cada iteração, a melhor fusão é selecionada através do algoritmo MBSA (*Matching Based Saving Algorithm*). Este critério pode ser considerado menos míope do que as usuais heurísticas de construção.

O método *das duas-fases* é baseado em métodos do tipo *agrupa-roteiriza* ou *roteiriza-agrupa*.

O método *agrupa-roteiriza* identifica antes o conjunto de clientes que será designado para cada veículo e então é computado o custo mínimo do problema do caixeiro viajante para a rota de cada veículo.

Gillet e Miller (1974), através da técnica *agrupa-roteiriza*, utilizaram um algoritmo de “varredura” para a fase 1 em que a localização dos clientes é representada em um sistema de coordenadas polares com a origem no depósito central. Um cliente é escolhido de forma aleatória e o raio da origem ao cliente é “varrido” no sentido horário ou anti-horário. Os clientes são designados aos veículos,

no decorrer da “varredura”, até atingir a restrição de capacidade. Depois um novo veículo é selecionado e a varredura continua até que a capacidade do novo veículo seja atingida e assim continua até que todos os clientes tenham sido “varridos” para um dado veículo.

Outras heurísticas que utilizam o método *agrupa-roteiriza* podem ser encontradas em Christofides *et al.* (1979) e Fisher e Jaikumar (1981). Neste último, a fase de agrupamento é resolvida otimamente através da utilização de um algoritmo rápido para o *problema de designação generalizada* (Fisher *et al.*, 1979²⁰).

O método *roteiriza-agrupa* constrói uma rota ótima para o problema do caixeiro viajante e depois particiona em rotas factíveis para o PRV (Beasley, 1983; Haimovich e Rinnooy Kan, 1985).

Um exemplo de heurística baseada em *otimização incompleta* é encontrado em Christofides *et al.* (1978). Este consiste essencialmente em um algoritmo *branch and bound*, transformado em uma heurística através da finalização prematura.

Segundo Fisher e Jaikumar (1981), as heurísticas que vêm sendo desenvolvidas para o PRV são modificações de heurísticas para o problema do caixeiro viajante. Assim como Osman (1993), eles classificam estas heurísticas de maneira similar à Christofides (1985), no entanto, apontam mais uma categoria: *métodos de melhorias*.

A maioria dos *métodos de melhorias* envolve a aplicação sucessiva de dois módulos: um método de construção, que produz uma solução factível inicial S de custo total $C(S)$; e uma técnica de melhoria iterativa que mantém a factibilidade e diminui gradativamente o custo da rota. Esta última envolve três conceitos fundamentais: um mecanismo de geração para alterar a solução inicial; estratégias de seleção das soluções alternativas e um critério de parada. Nestas heurísticas uma

²⁰ FISHER, M. L.; JAIKUMAR, L. N.; WASSENHOVE, V. (1979). *A multiplier adjustment method for the generalized assignment problem*. Philadelphia, University of Pennsylvania, The Wharton School, Department of Decision Science, [12:4] (Report 81-07-06).

dada solução é melhorada através de sucessivos procedimentos iterativos de troca de arestas. Procedimentos de troca para o PRV foram sugeridos por Christofides e Eilon (1969) e Russell (1977). Stewart e Golden (1984) usaram relaxação lagrangiana para transformar o múltiplo problema do caixeiro viajante em PRV e depois aplicaram um procedimento de troca de arcos similar a Lin e Kernighan (1973).

Metaheurísticas, tais como *busca tabu*, *algoritmos genéticos* e *simulated annealing*, podem ser vistas como métodos de melhorias (Gendreau *et al.*, 1994), embora alguns autores as reconheçam como uma categoria diferente dos métodos heurísticos tradicionais (Cunha, 1997; Souza, 1993).

Vários procedimentos eficientes de busca têm sido elaborados. Em especial, pesquisadores adaptaram idéias de outras áreas no desenvolvimento de *metaheurísticas*, ou técnicas que, superpondo-se a métodos heurísticos, guiam a busca com vista à superação da otimalidade local e à obtenção de soluções de melhor qualidade. As mais promissoras destas técnicas, na área de roteirização, incluem *simulated annealing* (SA), *algoritmos genéticos* (AG) e *busca tabu* (BT).

Simulated annealing, proposta por Kirkpatrick *et al.* (1983²¹), parte de conceitos de mecânica estatística, e é baseada em uma analogia com o processo de *recozimento* dos sólidos. A cura física refere-se ao processo de obtenção de estados de baixa energia de um sólido onde, a partir da substância derretida, passa-se a diminuir lentamente a temperatura, até atingir a do ponto de congelamento. Se o resfriamento for feito de forma lenta e apropriada, a configuração de mínima energia do sólido terá uma estrutura particular, como a observada em cristais. Caso contrário, o sólido resultante ficará congelado em uma estrutura localmente ótima, tal como um vidro ou um cristal com vários defeitos em sua estrutura.

Em analogia com problemas de otimização combinatorial, os diferentes estados da substância correspondem às diferentes soluções factíveis do problema, e a

²¹ KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. (1983). Optimization by Simulated Annealing. *Science*, n. 220, pp. 671-680.

energia do sistema, à função a ser minimizada. SA explora o espaço de soluções através da geração sequencial e aleatória de soluções candidatas. O mecanismo de geração geralmente consiste de uma prescrição simples para gerar a transição de uma solução a outra através de uma pequena perturbação.

As soluções que implicam na melhoria da função objetivo são aceitas incondicionalmente. Caso contrário, aplica-se um critério de aceitação estocástico. A probabilidade de aceitação da solução depende não só do valor da deterioração da função objetivo resultante, mas de um parâmetro T . O valor de T é gradualmente reduzido ao longo do processo e desempenha o mesmo papel que a temperatura em um sistema termodinâmico. Para altos valores de T , as configurações são equiprováveis e o algoritmo pode visitar praticamente qualquer uma delas. Com o decréscimo de T , reduz-se a probabilidade de aceitação de soluções deteriorantes (e portanto, o número de soluções acessíveis) até o momento em que o algoritmo atinge uma solução de baixo custo.

Simulated Annealing assegura que a probabilidade da solução piorar tende a zero com o crescimento do número de iterações. Este método foi aplicado ao PRV por Osman (1991²², 1993) *apud* Gendreau *et al.* (1994).

Algoritmos genéticos, introduzidos por Holland²³ (1975), procuram emular o fenômeno biológico da reprodução evolutiva. Ao contrário das outras metaheurísticas que exploram o espaço de soluções sequencialmente, estes algoritmos trabalham com populações de soluções, guiando a busca usando o “princípio da sobrevivência das mais aptas” (soluções de maior qualidade). A aptidão (*fitness*) de cada solução é medida por uma função equivalente à função objetivo, e a busca prossegue por um número de gerações onde a contribuição de cada indivíduo para a próxima geração é proporcional à sua aptidão. Isto é obtido selecionando-se

²² OSMAN, I. H. (1991). *Metastrategy simulated annealing and tabu search for combinatorial optimization problems*. London. Dissertation (Ph.D) - The Management School, Imperial College of Science and Medicine, University of London.

²³ HOLLAND, J. H. (1975). *“Adaptation in Natural and Artificial Systems”*. Ann Arbor, University of Michigan Press.

indivíduos aleatoriamente e usando uma função de probabilidade ponderada, onde os pesos refletem os valores reais de aptidão. Três operadores padrões são utilizados: *reprodução*, “*crossover*” (cruzamento) e *mutação*. A reprodução copia um indivíduo de uma geração para a próxima, cruzamento combina características de dois ou mais pais para produzir um ou mais filhos, e mutação realiza pequenas mudanças locais. Reprodução e cruzamento de indivíduos aptos são os mecanismos que impulsionam o melhoramento da função objetivo, enquanto a mutação mantém a diversidade da população e permite a exploração de novas regiões. Mais recentemente, a incorporação de busca local introduziu uma nova modalidade de algoritmos genéticos, os chamados *algoritmos meméticos*.

A *busca tabu* pode ser considerada uma técnica que incorpora conceitos selecionados de *inteligência artificial*. Seu objetivo é de emular usos inteligentes de memória, com o objetivo de cruzar fronteiras de factibilidade ou otimalidade local. Essencialmente, a BT é uma metaheurística ou procedimento heurístico para orientação da busca com vistas à obtenção de boas soluções em espaços de solução complexos. As regras que a compõem são suficientemente gerais para permitir sua utilização como dirigentes das operações de outros métodos heurísticos. Numerosos experimentos conduzidos para diferentes tipos de procedimentos têm mostrado que quando apropriadamente implementada, obtém resultados que freqüentemente igualam ou dominam técnicas previamente conhecidas. Juntamente com SA e AG, a busca tabu foi considerada pelo Comitê de Pesquisa Operacional (CONDOR, 1988²⁴) como “altamente promissora” para o tratamento de futuras aplicações práticas. O número de trabalhos descrevendo aplicações em BT na literatura recente revelam que esta previsão se concretizou. Abordagens puras e híbridas têm esclarecido novas marcas na obtenção de melhores soluções em problemas de roteirização de veículos (entre muitas outras áreas).

²⁴ CONDOR (1988). Comitê da Próxima Década de Pesquisa Operacional.

Uma das primeiras tentativas em aplicar busca tabu em PRV foi feita por Willard (1989²⁵). Em seu trabalho, o problema foi primeiro transformado em um problema do caixeiro viajante com a reprodução do depósito, e a busca foi restringida a soluções vizinhas que podem ser obtidas por meio de trocas *2-opt* (Lin, 1965) ou *3-opt* (Lin e Kernighan, 1973), satisfazendo as restrições do PRV.

Na heurística desenvolvida por Semet e Taillard (1993) para solucionar um PRV real contendo várias restrições, o movimento tabu básico consiste em mudar uma cidade de sua rota para uma outra alternativa.

Em Pureza e França (1991), a busca procede de uma solução para a próxima, trocando e inserindo vértices entre duas rotas e realizando movimentos *2-opt* em cada um. Osman (1993) usa uma combinação de movimentos *2-opt*, reindicação de vértices para diferentes rotas, e troca de vértices entre duas rotas.

Uma nova abordagem foi introduzida por Taillard (1993), na qual o conjunto de vértices é decomposto em subproblemas que podem ser resolvidos independentemente, conseguindo maior velocidade no método de busca iterativa. Resultados indicam que este método tem melhor desempenho que os outros. Na realidade, dois diferentes esquemas de decomposição foram propostos: um para problemas uniformes e outro para problemas não uniformes. A decomposição em regiões polares mostrou apresentar melhor performance no caso de problemas uniformes, onde o depósito está quase no centro e os vértices se encontram uniformemente distribuídos ao seu redor. Depois de uma pré-resolução dos subproblemas, vértices e rotas são trocados entre os subproblemas. Um método de decomposição baseado na arborescência do caminho mínimo partindo do depósito para todos os vértices é recomendado para problemas não uniformes, onde os vértices não estão regularmente distribuídos ao redor do depósito. Neste algoritmo, vértices e rotas não são transferidos entre as resoluções.

²⁵ WILLARD, J. A. G. (1989). *Vehicle Routing Using r-Optimal Tabu Search*. M.Sc. Dissertation. The Management School, Imperial College, London.

Em todos estes algoritmos, não é permitido que uma solução factível venha a ficar infactível, enquanto em Gendreau *et al.* (1994) são permitidos movimentos que resultam em soluções infactíveis quando são consideradas as restrições de comprimento e capacidade.

Barbarosoglu e Ozgur (1999) desenvolveram um algoritmo que usa a maioria dos princípios de busca tabu aplicados por Taillard (1993), mas propõem um novo procedimento de busca vizinha sem qualquer diversificação e um novo esquema de intensificação. A intensificação da busca começa em soluções que se suspeita estar no limite da região de ótimo local ou quiçá na região de ótimo global. Durante o estágio de melhoria da solução, o comportamento da função objetivo é examinado, e as soluções que pioram depois de uma série de melhorias são guardadas como um indicador do início de um vale em potencial ao redor do pico da função objetivo. A intensificação é executada ao redor destas soluções para explorar estas regiões mais de perto. Os autores decidiram não colocar o estágio de diversificação para diminuir custos computacionais.

Kelly e Xu (1999) desenvolveram uma heurística de busca tabu genérica para o PRV. Esta heurística consiste de duas etapas, na primeira das quais utilizou-se uma heurística simples para gerar um número suficiente de rotas distintas. Depois, na segunda fase, são identificadas as “boas” soluções, que são melhoradas usando um algoritmo para particionamento de números. Dado que este algoritmo é *NP-hard*, foi desenvolvida uma heurística de busca tabu para resolvê-lo de forma aproximada. Adicionalmente, novas rotas são criadas, combinando de forma inteligente rotas obtidas por técnicas de busca local. O método de busca local envolve o uso do algoritmo de busca tabu proposto por Taillard (1993).

As performances destas heurísticas foram comparadas através da aplicação de um conjunto de testes padrões encontrados na literatura (Christofides *et al.*, 1979). Os resultados apresentados por cada uma delas, quando aplicadas sobre os 14 problemas testes do *benchmark* de Christofides *et al.* (1979), podem ser comparados na tabela 2. Esta tabela foi adaptada de Gendreau *et al.*(1994) e as células em

destaque representam as melhores soluções encontradas na revisão bibliográfica realizada.

Os resultados mostram que todos os heurísticas tradicionais (de CW a AG, na tabela 2) apresentam soluções piores que os algoritmos que empregam técnicas de busca tabu e *simulated annealing*. No entanto, ao se analisar os resultados, deve-se tomar cuidado nas comparações parciais, pois a maioria das heurísticas (por exemplo, AG', OSA, OTS, T e GHL) foi executada várias vezes, levando em consideração diferentes valores de parâmetros, para a seleção da melhor solução.

Tabela 2: Comparação dos valores encontrados por 14 heurísticas alternativas.

problema	CW	GM	MJ	CMT1	CMT2	FJ	DV	AG	PF	T	OTS	OSA	OTS	GHL	KX
VRPNC1	585	532	575	547	534	524	586	556	536	524.61	524.61	528	524.61	524.61	524.61
VRPNC2	900	874	910	883	871	857	885	855	842	835.32	844	838.62	844	835.77	835.26
VRPNC3	886	851	882	851	851	833	889	860	851	828.98	835	829.18	835	829.45	826.14
VRPNC4	1204	1079	1259	1093	1064	1014	1133	1085	1081	1029.64	1044.35	1058	1044.35	1036.16	1028.42
VRPNC5	1540	1389	1545	1418	1386	1420	1424	1351	-	1300.89	1334.55	1378	1334.55	1322.65	1310.97
VRPNC6	619	560	599	565	560	560	593	577	560	555.43	555.43	555.43	555.43	555.43	555.43
VRPNC7	976	933	969	969	924	916	963	939	929	909.68	911	909.68	911	913.26	909.68
VRPNC8	973	888	999	915	885	885	914	913	887	865.94	866.75	866.75	866.75	865.94	865.94
VRPNC9	1426	1230	1289	1245	1217	1230	1292	1210	1227	1164.24	1184	1164.12	1184	1177.76	1171.33
VRPNC10	1800	1518	1770	1508	1509	1518	1559	1464	-	1403.21	1417.85	1417.85	1417.85	1418.51	1425.97
VRPNC11	1079	1266	1100	1066	1092	-	1058	1047	1049	1073.05	1042.11	1176	1042.11	1073.47	1042.11
VRPNC12	831	937	879	827	816	824	828	834	826	819.56	819.59	826	819.59	819.56	819.56
VRPNC13	1634	1770	1590	1612	1608	-	1562	1551	1631	1550.15	1547	1545.98	1547	1573.81	1581.98
VRPNC14	877	949	883	876	878	848	882	874	866	866.37	866.37	890	866.37	866.37	866.37

Fonte: Adaptada de Gendreau *et al.* (1994)

■ corresponde à melhor solução encontrada na literatura revisada.

CW: Clarke e Wright (1964) *savings algorithm*;

GM: Gillett e Miller (1974) *SWEEP algorithm*;

MJ: Mole e Jameson (1976) *generalized savings algorithm*;

CMT1: Christofides, Mingozzi e Toth (1979) *two-phase algorithm*;

CMT2: Christofides, Mingozzi e Toth (1979) *two-phase algorithm*;

FJ: Fisher e Jaikumar (1981) *two-phase algorithm*;

DV: Desrochers e Verhoog (1989) *MBSA algorithm*;

AG': Altinkemer e Gavish (1991) *PSA-T algorithm*;

PF: Pura e França (1991) *tabu search algorithm*;

T: Taillard (1992) *tabu search algorithm*;

OTS: Osman (1993) *tabu search algorithm*;

OSA: Osman (1993) *simulated annealing algorithm*;

GHL: Gendreau, Hertz e Laporte (1994) *tabu search heuristic*;

KX: Kelly e Xu (1999). A Set-Partitioning-Based Heuristic for the VRP.

3.3 ANÁLISE DE PERFORMANCE DE HEURÍSTICAS

Ball e Magazine (1981) notaram a significância de pesquisas em heurísticas de roteirização e programação, apontando como umas das tendências promissoras destas pesquisas, a análise de performance de heurísticas. A performance da heurística pode ser analisada através do “pior-caso”, probabilidades e análises estatísticas, complementando os estudos empíricos usuais. A comparação de heurísticas utilizando dados levantados em aplicações particulares, também é lembrada pelos autores que sugerem a comparação de diferentes heurísticas em banco de dados de problemas reais, através de um processo mais formalizado. A questão levantada é “o que, em particular, pode ser dito a respeito da performance da heurística aplicada no conjunto de dados levantados em um mesmo cenário no qual o problema se apresenta”.

Quando uma nova heurística é apresentada na literatura, seu desempenho deve ser cientificamente documentado e de uma maneira objetiva (Resende *et al.*, 1995). Uma das questões mais óbvias para qualquer heurística é como será seu desempenho. Dado que não há garantia de “otimalidade”, é claramente importante ter alguma estimativa de quão boa ou ruim será a solução apresentada por ela. É possível identificar, pelo menos, três métodos para responder esta questão: métodos analíticos (análise do pior-caso e análise probabilística ou desempenho médio), testes empíricos e inferência estatística (Reeves, 1993).

Cada método tem suas vantagens e desvantagens. Resumidamente, segundo Johnson & Papadimitriou (1985), análise probabilística (ou análise do desempenho médio) pode nos dizer muito, especialmente quando o algoritmo for aplicado sobre instâncias com características similares. No entanto, pela sua própria natureza, neste tipo de análise deve-se assumir uma dada distribuição de probabilidade para cada classe de instâncias. Se essa hipótese não for apropriada, o resultado da análise pode não estar ligado às instâncias manipuladas. Além disso, este tipo de análise não pode dizer nada definitivo a respeito do desempenho de um algoritmo sobre uma instância particular, já que uma instância particular pode ser muito atípica ou patológica.

Por outro lado, análise do pior caso pode assegurar garantias em instâncias individuais, e não envolve a suposição, implícita ou explícita, de qualquer distribuição de probabilidade. A desvantagem aqui é que, como a avaliação deve ser assegurada para todas as instâncias, até mesmo para uma que pode ser muito atípica ou patológica, pode existir uma discrepância considerável entre o comportamento de um algoritmo “na prática” e o seu melhor desempenho demonstrado. Uma forma de limitar esta desvantagem é aplicar a análise do pior caso sobre classes de instâncias que incluam aquelas que serão manipuladas, eliminando algumas das piores patologias.

Um problema final tanto com a análise do pior caso quanto com a análise probabilística, vem da natureza rigorosa destas duas técnicas. Analisar uma heurística através de uma destas duas técnicas pode ser uma tarefa matemática muito complexa. Heurísticas que possuem bons limites probabilísticos podem ser inapropriadas para análise de pior caso, e heurísticas que vem se mostrando boas no pior caso são, quase sempre, excessivamente difíceis de analisar probabilisticamente.

Análise empírica pode ser a mais apropriada se a escolha dos problemas testes incluir ou refletir instâncias reais dos problemas que serão manipulados. Ela pode ser, no entanto, errônea, se cuidados não forem tomados na escolha dos problemas testes ou se os problemas testes apresentarem características muito diferentes daqueles que serão manipulados na prática. Algumas vezes, as instâncias surgem em uma situação real, mas dados reais são escassos, e podem representar apenas uma pequena fração da população das possíveis instâncias, então problemas testes são frequentemente gerados por amostragem randômica daquilo que é assumido como sendo a população das instâncias de problemas de um determinado tipo (Reeves, 1993). O crescimento da Internet tem ajudado a promover a troca de informações e o compartilhamento de dados e códigos por pesquisadores. Por exemplo, há uma coleção de instâncias de toda uma variedade de problemas disponíveis por correspondência eletrônica, como descrito por Beasley (1990) e outros repositores *on-line* de problemas testes como TSPLIB, QAPLIB, MIPLIB, Netlib, etc.

4 SISTEMAS PARA ROTEIRIZAÇÃO DE VEÍCULOS

Cada problema tende a ser diferente e necessita de atenção particular. Nenhum algoritmo singular é apropriado para todas as situações, ou mesmo para um grande número delas. Assim, existe a necessidade de se desenvolver um método geral, que possa ser adaptado a cada situação específica (Rousseau, 1988). Dentro deste contexto, esse autor propõe um programa geral para sistemas comerciais, composto por quatro operadores básicos manipulando o conjunto de objetos (figura 4).

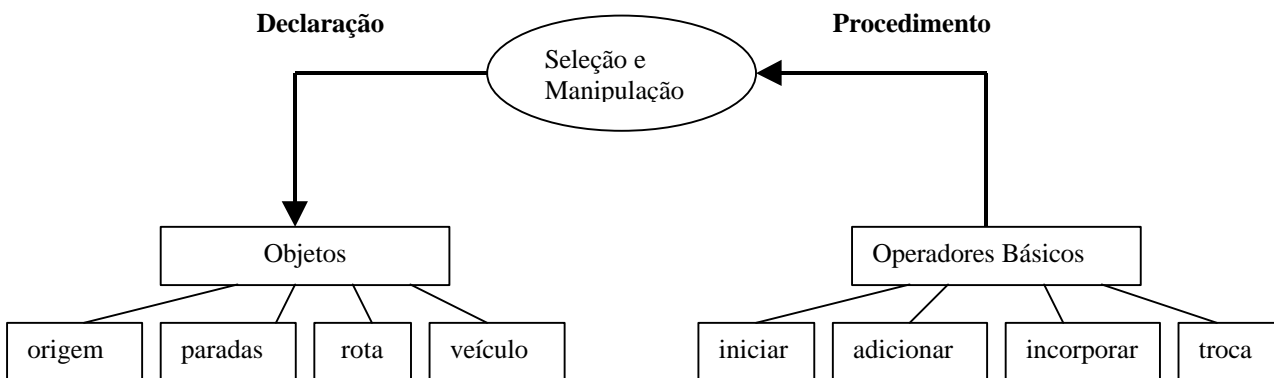


FIGURA 4 - OBJETOS E OPERADORES BÁSICOS

Fonte: Rousseau (1988)

Na figura 4 pode-se identificar três elementos:

- 1) *Objetos*. Os objetos possuem atributos que contêm informações para o operador agir sobre eles. Em particular, os atributos mais utilizados pelas heurísticas são:
 - Para a origem: coordenadas e número de veículos disponíveis;
 - Para as paradas: coordenadas, janela de tempo e demanda;

- Para a rota: número de paradas, comprimento, tempo de viagem e demanda total;
 - Para os veículos: velocidade, capacidade e distância máxima autorizada ou tempo de viagem.
- 2) *Operadores básicos*. Os objetos são manipulados por quatro operadores básicos, simbolizando fases distintas da heurística geral. Os operadores são:
- Iniciar: operador para criação e inicialização de novas rotas;
 - Adicionar: operador para a inserção de novas paradas na rota existente;
 - Incorporar: operador para junção das rotas;
 - Trocas: operador para a modificação da seqüência de paradas nas rotas.
- 3) *Um mecanismo de seleção*. Cada operador seleciona primeiro um ou mais objetos e depois os manipula. O processo de manipulação está intrinsecamente definido em cada operador, mas o processo de seleção é colocado sob controle do usuário de maneira que ele possa formular a seleção. Esta formulação associa valores com os objetos envolvidos na definição do problema.

Todos os operadores são independentes entre si e podem ser ativados pelo usuário na seqüência que ele desejar. A única restrição é óbvia: a primeira operação deve ser “iniciar”.

A maioria dos algoritmos heurísticos publicados, com todos as suas possíveis variações e combinações, pode ser facilmente implementada usando operadores e formulações para o mecanismo de seleção apropriado. Desta forma, o programa pode ser facilmente desenvolvido e através dele pode-se implementar um algoritmo apropriado para cada situação particular que possa aparecer.

4.1 CARACTERÍSTICAS DE SISTEMAS PARA ROTEIRIZAÇÃO

Assad (1988) relacionou as características típicas encontradas em *software* comerciais para roteirização de veículos (quadro 1), embora segundo o autor elas não estejam todas presentes, simultaneamente, em nenhum deles.

Quadro 1: Principais características de *software* comerciais

CARACTERÍSTICAS	diferentes tipos de veículos coletas e entregas, <i>backhauls</i> janelas de tempo tempo de carga/descarga variação da velocidade opção de contratação de serviços de terceiros
INFORMAÇÕES DO ROTEAMENTO	limite de capacidade do veículo duração máxima do tempo de percurso uso de horas extras horário de início e término do roteamento viagens de múltiplos dias, revezamento de motorista locais fixos para paradas múltiplos roteiros por veículos
OBJETIVOS	minimizar distância minimizar tempo de viagem minimizar número de veículos minimizar custo total
SAÍDAS	itinerário e programação do roteiro relatório de utilização do veículo relatório da jornada do motorista gráficos para roteiros superposições com rede viária localização e endereços

Fonte: Assad (1988)

Cunha (1997) sintetizou a visão de três autores no tocante aos principais requisitos e características de sistemas informatizados para roteirização de veículos. Tal resultado está apresentado no quadro 2.

Um importante atributo de sistemas para roteirização de veículos consiste na precisão das informações geográficas sobre as localizações dos clientes e redes viárias. Geralmente, a localização dos clientes é disponibilizada na forma de um endereço, localizado com o auxílio de um mapa da região. Uma maneira simples de manipular estes dados é atribuir coordenadas x,y ou latitude e longitude aos clientes.

A distância entre os clientes é computada achando-se o menor caminho de um cliente para todos os outros.

Quadro 2: Síntese dos principais requisitos e características de sistemas para roteirização de veículos segundo alguns autores.

CARACTERÍSTICA	ASSAD (1988)	RONEN (1988)	BODIN (1990)
Recursos, restrições e condicionantes			
Roda em microcomputador, porém com interface para <i>mainframe</i>	-	X	-
Uma ou múltiplas bases	X	X	X
Diferentes tipos de veículos	X	-	X
Coletas e entregas – “backhauls”	X	X	X
Janelas de tempo	X	-	-
Tempos de carga e descarga	X	-	-
Velocidades variáveis	X	X	-
Contratação de terceiros	X	X	-
Limite de peso e volume	X	X	-
Múltiplos compartimentos por veículos	-	X	-
Duração máxima do roteiro	X	X	X
Contabilização de horas extras	X	-	X
Horários de início e término de viagem	X	-	-
Roteiros com duração superior a um dia; pernoite; revezamento de motoristas	X	X	-
Locais de parada fixos (ex. almoço)	X	-	-
Restrições quanto ao tamanho do veículo e seus equipamentos para um cliente	X	-	X
Zonas de entregas e possibilidade de fracionamento de carga; roteiros fixos	X	-	-
Sistema de georeferências; barreiras físicas e restrições de circulação de veículos	X	X	-
Múltiplos roteiros por veículo	X	-	-
Função Objetivo			
Minimizar distância	X	X	-
Minimizar tempo de viagem	X	X	-
Minimizar número de veículos	X	X	-
Minimizar custo total	X	X	X
Resultados			
Roteiro e programação de cada veículo	X	-	-
Relatório de utilização do veículo	X	-	-
Relatório de programação do motorista	X	-	-
Roteiros gráficos	X	X	-
Relatórios definidos pelo usuário	-	X	-
Alteração manual de soluções	-	X	-

Fonte: Cunha (1997)

Assad (1988) destaca a importância do uso de Sistemas de Informações Geográficas em roteirização em função da precisão com que estes sistemas calculam a matriz do tempo de viagem e distância entre clientes. Em princípio, isto pode ser

imediatamente disponibilizado através de um banco de dados geográficos de uma rede viária. Um SIG possibilita a localização automática de clientes e endereços, bem como a determinação da distância e tempo de viagem entre pontos de atendimento através da aplicação de algoritmos de caminho mínimo (mínimo tempo ou mínima distância ou uma ponderação de ambos) sobre a rede que representa a malha viária da região de interesse. Mesmo que o número de clientes seja relativamente pequeno, o número de nós envolvidos na rede pode ser grande, principalmente no caso de roteirização urbana, onde opera-se sobre uma rede detalhada de ruas e/ou segmentos de ruas. Assim, o autor aponta duas razões para utilizar SIG's em roteirização:

- 1) Exatidão na estimativa do tempo de viagem: subestimar o tempo de viagem conduz à faltas no prazo final, enquanto superestimar pode acarretar em subutilização do veículo e motorista, criando tempo improdutivo.
- 2) Melhor qualidade de solução: associada a configuração da rota. Quanto mais detalhada for a rede, no sentido de cobertura de todos os nós e *links* (que representam as paradas e a ligação entre estas paradas), mais detalhadas serão as informações com relação ao caminho e, conseqüentemente, maior será a precisão no cálculo do tempo de viagem, passado para o algoritmo de roteirização.

Os SIG's são especialmente convenientes quando os clientes variam muito e os locais de entrega e/ou coleta estão em constante mudança no sistema de distribuição, como no caso de entrega domiciliar em geral (por exemplo, lojas de departamento).

Na tabela 3 foram relacionadas as principais características encontradas em alguns sistemas para roteirização que são SIG's.

Tabela 3: Alguns sistemas para roteirização que são SIG e suas principais característica

Produto	GeoRoute	GeoRoute 5	Load Manager	LoadExpress Plus	Manugistics Routing & Scheduling
Publicação	Kositzky & Associates, Inc.	GIRO Enterprises, Inc.	Roadnet Technologies, Inc.	Information Software, Inc.	Manugistics, Inc.
Sistema operacional					
Windows 95	Sim	Sim	Sim	Sim	Sim
Windows NT	Sim	Sim	Sim	Sim	Sim
Unix	Não	Não	Não	Não	Não
Tamanho do problema resolvido pelo sistema					
Número de paradas	4600	Ilimitado	s/ atribuição	Ilimitado	Ilimitado
Número de veículos	512	Ilimitado	s/ atribuição	500	Ilimitado
Número de depósitos	256	Ilimitado	s/ atribuição	Ilimitado	Ilimitado
Roteirização					
Tempo real	Não	Não	Não	Não	Sim
Diária	Sim	Sim	Não	Sim	Sim
Planejamento	Sim	Sim	Não	Sim	Sim
Interface	GeoWhiz	ArcInfo MapInfo		Patenteada/ pode trabalhar c/ dados de todos os SIG's	Xeta, Rockwell, Cadec, Auroach
Características					
Atribui motorista à rota	Sim	Não	Não	Sim	Sim
Instrução de cada rota	Sim	Sim	Não	Sim	Sim
Manifesto de carga	Não	Não	Sim	Sim	Sim
Outras características	Funciona p/ roteirização urbana e interurbana; permite multi-depósito e redespacho do veículo	Suporta operações em nós, arcos e mistas		Constrói rotas otimizadas; entregas programadas e analisa padrões de distribuição	Gerenciamento de recursos; permite gerenciar programação de motoristas e veículos e requisitos de equipamento para carregamento
Tipo de aplicação	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.

Tabela 3: Alguns sistemas para roteirização que são SIG e suas principais característica (Continuação)

Produto	OVERS	RIMMS	ROADNET 5000	RoadShow for Windows	RoutePro
Publicação	Bender Management Consultants	Lightstone Group, Inc.	Roadnet Technologies, Inc.	RoadShow International, Inc.	CAPS LOGISTIC
Sistema operacional					
Windows 95	Sim	Sim	Sim	Sim	Sim
Windows NT	Sim	Sim	Sim	Sim	Sim
Unix	Sim	Não	Não	Não	Não
Tamanho do problema resolvido pelo sistema					
Número de paradas	10.000	Ilimitado	Ilimitado	8.000	HW-based
Número de veículos	1.000	Ilimitado	Ilimitado	Ilimitado	HW-based
Número de depósitos	100	Ilimitado	Ilimitado	Ilimitado	HW-based
Roteirização					
Tempo real	Sim	Não	Não	Sim	Sim
Diária	Sim	Sim	Sim	Sim	Sim
Planejamento	Sim	Não	Não	Sim	Sim
Interface	Map Objects	Arquivos formato ESRI	GDT Maps	GDT, Etak, MapInfo, e software patentado	Etak, GDT, PCMIler, Horizons Technology
Características					
Atribui motorista à rota	Sim	Sim	Sim	Sim	Sim
Instrução de cada rota	Sim	Sim	Sim	Sim	Sim
Manifesto de carga	Sim	Não	Não	Sim	Sim
Outras características	Pode otimizar rotas em múltiplos períodos de tempo, respeitando restrições de tempo/espço - otimiza número e localização de terminais e áreas de serviços			Calcula custos das soluções baseado em informações fornecidas pelo usuário	Pode ser customizado
Tipo de aplicação	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de passageiros; - transporte de cargas; - coletas e entregas.

Tabela 3: Alguns sistemas para roteirização que são SIG e suas principais característica (Continuação)

Produto	RouteSmart Neighborhood	RouteSmart Point-to-Point	Routronics 2000	SHIP CONS II	Taylor II
Publicação	RouteSmart Technologies	RouteSmart Technologies, Inc.	Carrier Logistics	Insight, Inc.	F&H Simulation, Inc.
Sistema operacional					
Windows 95	Sim	Sim	Não	Sim	Sim
Windows NT	Sim	Sim	Não	Sim	Sim
Unix	Sim	Sim	Sim	Não	Não
Tamanho do problema resolvido pelo sistema					
Número de paradas	Ilimitado	Ilimitado	Ilimitado	Ilimitado	1.000
Número de veículos	Ilimitado	Ilimitado	Ilimitado	Ilimitado	100
Número de depósitos	1 + intermediários	1 + intermediários	Ilimitado	Ilimitado	1.000
Roteirização					
Tempo real	Não	Não	Não	Não	Não
Diária	Não	Sim	Sim	Sim	Não
Planejamento	Sim	Sim	Não	Sim	Sim
Interface	GIS Plus (DOS), ArcInfo	ArcView 3.0	MapInfo	GDT, MapInfo, Etak	
Características					
Atribui motorista à rota	Não	Não	Não	Não	Não
Instrução de cada rota	Sim	Sim	-	Não	Não
Manifesto de carga	Não	Não	Sim	Sim	Não
Outras características	Sistema de leitura de distância, manipula caminhos, motoristas e combina rotas		Roteirização completa dos clientes e interface com sistemas de comunicação sem fio	Custo baseado em otimização inteira; usuário configura a tela	Animação em 2-D e 3-D; módulo de estatística avançada
Tipo de aplicação	Nenhum dos citados.	- Transporte de cargas; - coletas e entregas.	- Transporte de cargas; - coletas e entregas.	- Transporte de passageiros; - transporte de cargas; - coletas e entregas.	- Transporte de passageiros; - transporte de cargas; - coletas e entregas.

Tabela 3: Alguns sistemas para roteirização que são SIG e suas principais característica (Continuação)

Produto	Territory Planner	TESYS	TransCAD	Trapeze-FX, Para, CI, FL	TruckStops for Windows
Publicação	Roadnet Technologies	Inform Software Corporation	Caliper Corporation	Trapeze Software Group	Micro Analytics, Inc.
Sistema operacional					
Windows 95	Sim	Não	Sim	Sim	Sim
Windows NT	Sim	Sim	Sim	Sim	Sim
Unix	Não	Sim	Não	Não	Não
Tamanho do problema resolvido pelo sistema					
Número de paradas	Ilimitado	3.000	Ilimitado	Ilimitado	+16.000
Número de veículos	Ilimitado	1.000	Ilimitado	Ilimitado	+16.000
Número de depósitos	Ilimitado	500	Ilimitado	Ilimitado	+16.000
Roteirização					
Tempo real	Não	Sim	Sim	Sim	Não
Diária	Não	Sim	Sim	Sim	Sim
Planejamento	Sim	Não	Sim	Sim	Sim
Interface	GDT Maps	CDPD, GPS, RF	Pode trabalhar com dados de todos SIG, Maptitude, GIST	Etak, MapInfo, TIGER, ArcInfo, Navtech	Qualquer produto capaz de carregar dados via ASCII.
Características					
Atribui motorista à rota	Sim	-	Não	Sim	Sim
Instrução de cada rota	Sim	Sim	Sim	Sim	Não
Manifesto de carga	Não	Sim	Não	Sim	Não
Outras características			Kit de ferramentas incluindo problemas de transporte, e vários métodos para otimização de localização	Conceito multi-modal / inter-modal; estrutura de dados OBDC	
Tipo de aplicação	- Transporte de cargas; - coletas e entregas.	- Transporte de passageiros; - transporte de cargas; - coletas e entregas.	- Transporte de passageiros; - transporte de cargas; - entregas e coletas.	- Transporte de passageiros.	- Transporte de cargas; - coletas e entregas.

Fonte: http://lionhrtpub.com/orms/surveys/Vehicle_Routing/Vehicle_Routing_Survey.html

5 METODOLOGIA

5.1 EXPERIMENTAÇÃO EM OTIMIZAÇÃO

Segundo Resende *et al.* (1995), no passado recente, a necessidade de uma ciência empírica rigorosa dos algoritmos vem sendo muito enfatizada (Golden *et al.*, 1986; McGeoch, 1986¹; Bland *et al.*, 1993; Hooker, 1994). Testando uma heurística através de uma grande variedade de problemas pode-se ter uma idéia do quão bom será seu desempenho em geral, e em quais circunstâncias ela será relativamente boa ou ruim.

Experimentação em otimização envolve tipicamente a aplicação de um algoritmo sobre uma ou mais classes de problemas e a posterior análise empírica dos fatores que causaram impactos em seu desempenho. Os principais fatores apontados por Golden *et al.* (1986), incluem: a configuração do *hardware* (isto é, o modelo específico do computador, sistema operacional e características especiais, tais como o número de coprocessador); implementação do *software*; natureza do algoritmo (grau de otimalidade requerido); tipo de problema (origem, tamanho e complexidade dos problemas testes). Entre estes fatores, aquele relativo a seleção dos problemas testes requer atenção especial.

Golden e Stewart (1985) interpretam análise empírica como análise originada de, ou baseada em, experiência computacional. Um experimento consiste em um

¹ McGEOCH, C. (1986). *Experimental Analysis of Algorithms*. Ph.D. thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

conjunto de testes processados sob condições controladas para um propósito específico: demonstrar um fato, checar a validade de uma hipótese, ou examinar a performance de alguma coisa nova.

De maneira semelhante, para Resende *et al.* (1995), em testes computacionais de um algoritmo, o experimento consiste na resolução de uma série de instâncias utilizando uma implementação computacional. Durante este processo, o experimentador tem uma grande variedade de decisões a serem tomadas na escolha dos fatores a serem explorados, que irão influenciar nas saídas ou resultados do experimento. Os autores ainda diferenciam o experimento em: (a) experimento comparativo, onde compara-se a performance de diferentes algoritmos, aplicados sobre uma mesma classe de problemas; ou (b) descritivo, para caracterizar ou descrever a performance de um algoritmo isoladamente. Como estes experimentos diferem em sua meta principal a ser alcançada, os problemas testes de importância também diferem.

Neste estudo foi feita uma análise empírica comparativa entre os *software Delivery* e *TransCAD*, através dos resultados gerados pelos seus algoritmos em um conjunto de problemas testes, visando analisar o desempenho do algoritmo do *software Delivery*. A opção por um experimento comparativo e não descritivo é justificado pelas seguintes limitações com relação à interface do *software Delivery*:

- 1) A base cartográfica inserida no sistema é única (que constitui uma base de dados cartográficos da malha viária do Brasil) e foi desenvolvida pela própria empresa, que defende a idéia e diz ter a política de não fornecer, em hipótese alguma, esta base para implementação em outros sistemas (que não o dela) que tenham flexibilidade de importação.
- 2) Este fator ainda é agravado pela impossibilidade de manipular o *Delivery* sem utilizar a base cartográfica, ou seja, é impossível inserir localidades (depósito e paradas) que não estejam relacionadas à uma cidade, em uma determinada Unidade Federal (UF) da base cartográfica do Brasil.

Em princípio, estas limitações inviabilizaram o desenvolvimento de uma metodologia para testar empiricamente o *software*, pois não há como comparar os resultados gerados pelo *Delivery* com nenhuma outra ferramenta (porque elas nunca teriam a mesma base, e portanto não seria possível reproduzir os mesmos testes nos dois sistemas) e também não daria para utilizar como referência uma base de dados de testes padrão (como, por exemplo, os problemas clássicos de roteirização de veículos com janela de tempo propostos por Solomon, 1987) devido a incompatibilidade de entrada dos dados no *Delivery*.

Sendo assim, esforços foram feitos pela empresa que criou o *software* no sentido de preparar o sistema, de maneira que se pudesse desenvolver uma metodologia bem estruturada conceitualmente e que tivesse aplicação prática. O *software Delivery* foi preparado de forma que se pudesse inserir uma matriz de distância entre as localidades (depósito e paradas). Esta matriz pode ser gerada, por exemplo, no *software TransCAD*, utilizando as coordenadas x,y dos pontos (ou latitude e longitude) como dados de entrada. Gerada a matriz distância desta forma, ela representará os dados de entrada no *software Delivery*. Visto que será necessário o tratamento dos dados de entrada com o auxílio do *software TransCAD*, é conveniente utilizá-los também na rotina de roteirização como objeto de comparação.

Para que se pudesse realizar uma análise mais robusta, também foi providenciada a implementação, no *Delivery*, de uma base de dados cartográficos comum ao *software TransCAD*, de forma que o comportamento dos algoritmos seja analisado em situações reais, onde espera-se que uma rede viária real contribua para aumentar a complexidade do problema. A base cartográfica comum nos *software Delivery* e *TransCAD* será a malha viária da cidade de São Carlos/SP criada pelos alunos de pós-graduação do Departamento de Transportes, da Escola de Engenharia de São Carlos (Silva *et al.*, 1996), que a empresa concordou em inserir no *Delivery*.

Agora que já se conhece as reais limitações para este estudo, é possível propor uma metodologia, como se descreve a seguir.

5.2 ESCOLHAS DAS CLASSES DE PROBLEMAS PARA TESTES

O estagio inicial do experimento consistiu na escolha das classes de problemas a serem testadas e os problemas testes que irão representá-las. Esta etapa pode ser considerada a mais importante do processo, visto que ela conta com uma série de tomadas de decisões com relação aos fatores ou características a serem explorados na avaliação. Abaixo encontram-se os passos seguidos neste processo:

- 1) Foram realizadas visitas em três tipos de empresas padrão (uma distribuidora de bebidas, uma transportadora e uma distribuidora de alimentos) envolvidas com o problema de roteirização, visando caracterizar o problema real e através dele levantar o perfil teórico em que se enquadram, dando suporte na escolha da classe de problemas a ser testada. A melhor maneira encontrada para atingir este objetivo foi analisar qualitativamente os sistemas de roteirização utilizados por elas, bem como a maneira como eles são operados. Conclusões foram tiradas com relação aos tipos de problemas encontrados em situações reais e as características que são relevantes e desejadas em sistemas desta natureza.
- 2) Os *software Delivery* e *TransCAD* foram analisados qualitativamente e caracterizados segundo o esquema de classificação proposto por Desrochers *et al.* (1990), representando mais uma diretriz no processo de escolha das classes de problemas a serem testadas, uma vez que auxilia na identificação de todas as classes possíveis do problema que podem ser resolvidas pelas duas ferramentas. Ou seja, antes de partir para a análise empírica, é conveniente realizar uma análise qualitativa das características presentes nos sistemas, visando os pontos em comum. Aqui, também foi possível identificar o funcionamento destes sistemas (em nível operacional) para reconhecimento das limitações e incompatibilidades com relação às entradas de dados.
- 3) Realizou-se uma pesquisa bibliográfica para levantamento dos problemas testes clássicos que representam classes específicas do problema de roteirização de veículos e são utilizados pela comunidade acadêmica

como referencial na análise de performance de novas heurísticas propostas para resolver tal problema.

- 4) Testes computacionais: resolução dos problemas testes utilizando os *software Delivery* e *TransCAD*.

5.3 ESCOLHA DAS MEDIDAS DE PERFORMANCE

As medidas de performance tendem a ser agrupadas em três áreas: qualidade da solução, esforço computacional e robustez.

Qualidade da solução

Geralmente ela é medida pelo desvio, para a solução ótima, da solução apresentada pelo algoritmo testado. Quando a solução ótima é desconhecida, os algoritmos são testados em relação aos resultados de seu melhor “concorrente”, fornecendo um limitante inferior (ou superior) que constitui uma medida eficiente da qualidade da solução.

Quando se trata de problemas testes reais, uma boa medida de performance consiste em comparar os resultados gerados pelo sistema com aqueles obtidos na roteirização real.

Esforço computacional

McGeogh (1995) notou que uma previsão segura do tempo de processamento em função de diferentes medidas de desempenho, sobre um amplo espectro de sistemas computacionais, permanece um problema aberto. Ela apresenta uma lista de sugestões para achar boas medidas de desempenho. Geralmente, o esforço computacional é estimado pelo tempo total de processamento em função da estrutura e tamanho do problema.

Os tempos de processamento dos problemas testes encontrados na literatura, para diferentes aplicações heurísticas, não constituem um referencial para comparação eficiente devido às diferenças de especificação de *hardware*, o que representa uma discrepância muito grotesca para efeito de comparação.

Uma alternativa para o caso aqui estudado seria comparar os tempos de processamento dos *software Delivery* e *TransCAD*, em função do tamanho e estrutura do problema. Ou seja, para todos os testes executados no *Delivery* serão anotados os tempos de processamento. Os mesmos testes serão executados no *TransCAD*, no mesmo equipamento, e os tempos de processamento também serão anotados. A partir destes resultados será feita uma análise comparativa em função da estrutura e tamanho do problema.

Plotando-se os gráficos dos tempos de processamento em função do tamanho e estrutura dos problemas, pretendeu-se fazer uma estimativa da complexidade computacional envolvida em cada sistema.

Robustez

Geralmente, a robustez é baseada na habilidade de uma heurística executar bem uma grande variedade de problemas testes. Uma heurística que obtém um excelente resultado apenas para uma instância do problema não é robusta.

A robustez deve ser medida ajustando-se adequadamente a heurística para resolver bem uma única instância do problema. A estratégia de resolução da heurística e os parâmetros devem ser mantidos constantes para um conjunto de problemas testes. Onde são escolhidos os valores dos parâmetros, pequenas mudanças devem ser feitas de forma que alguma medida da sensibilidade na execução da heurísticas possa indicar a robustez do algoritmo. No mínimo, para cada classe de problema, o efeito da variação no seu tamanho deve ser analisado.

Neste experimento, a robustez foi analisada em função da classe e tamanho do problema. Foi escolhida entre as heurísticas, através de técnicas estatísticas, aquela que apresentava a menor sensibilidade (menor desvio) na qualidade da solução em função da classe e tamanho do problema, como sendo a mais robusta.

Ambiente de teste

Os mesmos problemas testes foram executados pelos *software* no mesmo computador (igual configuração). Os resultados em termos de tempo de processamento e qualidade da solução achados para cada instância do problema devem ser diretamente comparáveis entre as duas heurísticas. Assim, foram controlados os fatores relacionados à *hardware* – modelo, disponibilidade de memória, velocidade do CPU, número de processadores; e *software* – sistema operacional.

5.4 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS

As heurísticas dos sistemas foram comparadas através de métodos estatísticos propostos por Golden e Stewart (1985). Estas técnicas pressupõem que os problemas testes formem uma amostra representativa da classe de problemas abordada. Embora, segundo os autores, isto seja impossível de avaliar, esta suposição sublinha a importância que começa na escolha do próprio conjunto de problemas testes. Parece haver um forte consenso de que certos *benchmarks* de problemas testes são apropriados para trabalhar, como é o caso dos que foram utilizados.

Uma das técnicas consiste em um teste estatístico não paramétrico, designado Teste de Wilcoxon, para comparar duas heurísticas. O procedimento consiste, inicialmente, na aplicação das duas heurísticas (no caso, heurística do *Delivery* – HD; heurística do *TransCAD* – HT) sobre os problemas testes de roteirização de veículos, onde os limites inferiores são conhecidos.

O teste Wilcoxon utiliza grau de diferenças “locais” para avaliar duas populações. No contexto de comparação de heurísticas, este teste é usado para comparar duas heurísticas de cada vez, baseado em um conjunto de n problemas de vários tamanhos e estruturas. Os parâmetros x_i e y_i correspondem às porcentagens, acima do limite inferior, geradas pelas duas heurísticas para o problema i . O fator estatístico Wilcoxon (W) é computado da seguinte forma:

- 1) Calcula-se a diferença $d_i = x_i - y_i$;
- 2) Ordena-se os valores absolutos de d_i em ordem crescente de 1 a n . Se algum $d_i = 0$, este não é considerado e diminui-se 1 de n . Se ocorrerem empates nas diferenças absolutas, calcula-se a média dos casos empatados e utiliza-se esta média em todos os itens em que ocorreu o empate;
- 3) Atribui-se o sinal de d_i aos valores do *rank* e denota-se este número por R_i ;
- 4) Finalmente, obtém-se a soma $W = R_1 + R_2 + \dots + R_n$.

A hipótese nula é que $E[HD] = E[HT]$, e deverá ser rejeitada, com nível de significância α , se:

$$W > W_{1-\alpha/2} \text{ ou } W < W_{\alpha/2}$$

$$W > W_{1-\alpha} \text{ ou}$$

$$W < W_{\alpha}$$

dependendo da hipótese alternativa ser:

$$E[HD] \neq E[HT],$$

$$E[HD] > E[HT], \text{ ou}$$

$$E[HD] < E[HT].$$

Valores críticos do fator estatístico Wilcoxon podem ser encontrados em tabelas de vários livros de estatística não paramétrica. Além disso, para $n \geq 10$, o valor crítico de W_{α} pode ser aproximado por:

$$W\alpha = Z(\alpha) \sqrt{n(n+1)(2n+1)/6}$$

onde $Z(\alpha)$ é o desvio padrão normal, tal que uma proporção α da área está à esquerda de $Z(\alpha)$.

6 ANÁLISE QUALITATIVA DE *SOFTWARE* PARA ROTEIRIZAÇÃO DE VEÍCULOS

Neste capítulo foi realizada uma análise qualitativa de alguns *software* para roteirização de veículos disponíveis no mercado. Em uma primeira etapa foi realizada uma visita a três empresas envolvidas com roteirização. Ferramentas utilizadas por elas para auxiliar a atividade de roteirização, foram, na medida do possível, especificadas. Conclusões foram tiradas com relação às características mais relevantes para problemas de roteirização reais, primeiro fator que influenciou na escolha das classes de problemas a serem testadas. Em uma segunda etapa foi feita uma descrição detalhada das características dos *software Delivery* e *TransCAD*, principais objetos de análise quantitativa nesta pesquisa, para levantamento das características comuns possíveis de serem consideradas durante o procedimento para roteirização de veículos, segundo fator considerado na escolha das classes de problemas testadas.

6.1 VISITAS À EMPRESAS

6.1.1 Transportadora

A primeira empresa visitada consiste em uma grande transportadora de produtos em geral. Esta empresa é formada por um centro de distribuição, estrategicamente localizado em Campinas/SP, e mais dezesseis filiais distribuídas por todo o país.

Sua logística operacional baseia-se no conceito Pólo/Central de transbordo, ou seja, toda mercadoria coletada nos pólos ou filiais da empresa é reunida no centro de distribuição de Campinas e depois de separada, é enviada ao seu destino final. Assim, a operação de coleta é realizada separadamente da operação de entrega. Geralmente, a coleta corresponde a um par origem-destino, pois um único cliente é capaz de lotar um veículo e sua carga é fracionada no centro de distribuição que terá destino em vários pontos distribuídos pelo país. Portanto, para a operação de coleta há muita demanda repetida (clientes não variam muito), enquanto que para a operação de entrega os clientes mudam todos os dias. É possível que um veículo, após realizar a atividade de distribuição, termine o roteamento em uma base diferente daquela que saiu.

A empresa opera uma frota de veículos de vários tamanhos e capacidades (caminhões grandes, médios e pequenos, vans e carros), sendo que 95% deles são terceirizados. Existe a diferenciação dos veículos destinados à coleta e entrega. A frota está distribuída pelas filiais, e a quantidade de veículos em cada terminal varia de acordo com a demanda. Os veículos possuem diferentes características e sua utilização vai depender da compatibilidade com o produto a ser transportado. Muitos produtos precisam de auxílio de equipamentos especiais durante o carregamento e descarregamento do veículo.

Como a maioria das coletas e entregas são realizadas em estabelecimentos comerciais, a roteirização deve respeitar o horário de funcionamento dos mesmos. Apesar disso, a empresa se compromete em entregar qualquer tipo de mercadoria em qualquer parte do país em, no máximo, 24 horas.

Assim, as informações ou características do problema que são relevantes para a realização do roteamento são: realiza coletas ou entregas, não há prioridade de atendimento de clientes, roteirização urbana e interurbana, restrição de mercadoria/veículo, considera janela de tempo e prazo máximo para entrega, veículos de diferentes características e capacidades, tempos de carga e descarga, múltiplos depósitos.

Software utilizado pela empresa: TruckStops

O *TruckStops* consiste em uma ferramenta SIG desenvolvida há cerca de 15 anos pela empresa MicroAnalytics Inc., tendo como representante no Brasil a T&I. Este sistema é utilizado por grandes empresas, tais como Sadia, Kibon, Yopa, Danone, Philip Morris e Ceval.

Os pontos avaliados no *software* foram:

- *Requisitos mínimos de hardware para instalação:* o sistema opera com micro IBM-PC 486 DX2/66 Mhz ou acima
- *Ambiente de instalação:* sistema operacional MS-DOS (desde 1996 já existe a versão Windows, mas tudo aqui se refere apenas à versão DOS)
- *Tipo de instalação:* local (*stand alone*)
- *Custo da versão:* US\$ 17 mil (o *software* não possui módulos, ou seja, este preço corresponde à versão completa e pronta para utilização)
- *Tempo para implantação:* 6 meses
- *Tempo em operação:* 7 anos
- *Suporte operacional:* existem 3 versões do *software* na empresa e 5 funcionários envolvidos para suporte operacional (2 turnos de trabalho)
- *Versão para avaliação antes da aquisição do software:* foi disponibilizada uma versão do *software* durante dois meses (acompanhada do manual) para avaliação.
- *Decisão de compra:* na época em que foi adquirido o produto (há 7 anos atrás), o “*TruckStops*” era uma das poucas opções disponível no mercado, era o mais barato e o mais utilizado nos EUA
- *Consultoria e Treinamento:* não foi dada consultoria e nem treinamento básico aos funcionários que iriam operar o sistema
- *Tempo necessário para a utilização de toda a capacidade do software (do ponto de vista do usuário):* segundo o funcionário envolvido na atividade operacional que utiliza o *software*, foi necessário um mês para ler o manual e aprender o procedimento para manipulação do mesmo

- *Manual*: em português, e traz todas as telas do sistema. Segundo o usuário, o manual é desorganizado e pouco didático

Base Cartográfica

O “*TruckStops*” não traz uma base cartográfica pronta. O usuário deve digitalizar a base cartográfica (caso ele deseje trabalhar com uma), através de um editor de mapas do *software*. No editor de mapas são dadas as coordenadas de início e fim das retas que irão representar uma dada rodovia ou parte dela (no caso de mapas interurbanos); ou das ruas (para o caso de mapas de cidades). Este editor (tela na figura 5) gera um arquivo do tipo .lmk.

Este sistema se mostra bastante flexível no sentido de que é possível cadastrar e adicionar quantos locais e ligações entre locais forem desejados.

Outra possibilidade é trabalhar com uma base cartográfica no formato MapInfo. Esta base cartográfica serviria apenas como um pano de fundo para visualização do usuário, pois as rotas são calculadas pelas distâncias euclidianas ou distâncias cadastradas entre os pontos de origem e destino. A desvantagem de se trabalhar em conjunto com esta base cartográfica é que o sistema fica muito lento, tanto para processamento como para visualização (a cada “zoom” o mapa é redesenhado).

TRUCK - EDIT	STATE _____
NAME / LOC:	
FOR LANDMARK TEXT	FOR LINE LANDMARK
	START COORDINATES
TEXT	X:
TEXT X:	Y:
TEXT Y:	
	END COORDINATES
	X:
TYPE OF POINT LINE	Y:
POINT / LINE: (L ou P)	
PRIORITY: (cor)	

FIGURA 5 - TELA DO EDITOR DE MAPAS DO *TRUCKSTOPS*

Existe um cadastro onde são colocadas as barreiras (lagos, parques, ou seja, barreiras físicas em geral). As distâncias entre os locais são calculadas em função das suas coordenadas (x,y), ou seja, distâncias euclidianas. É possível introduzir um parâmetro de ajuste para que esta distância se aproxime da distância real.

Existe também um cadastro (CADASTRO DE DISTÂNCIA) onde pode ser colocada a distância real entre localidades. Se houver alguma distância cadastrada o sistema a adota para fazer o cálculo; se não houver nada cadastrado, a distância é calculada em função das coordenadas, utilizando a métrica euclidiana (e alterada em função do parâmetro de ajuste, se for o caso).

Entrada de Dados

Importação dos dados: para realizar a importação dos dados é necessário seguir um formato específico para criar um arquivo tipo texto (a descrição deste formato encontra-se disponível no manual). Não possui interface com outros banco de dados ou planilhas eletrônicas.

Cadastros: As entidades, características e restrições levadas em consideração no roteamento são especificadas e atribuídas através de quatro cadastros:

CADASTRO DE PARADAS. Os campos dos dados são: nome da localidade; ID (existem dois campos para identificação, que podem ser CGC, código, etc.); endereços (também podem ser cadastrados dois endereços por localidade – rua, número, bairro, cidade, estado, telefone); coordenadas (x,y), tempo fixo de parada, Janela de Tempo para atendimento, quantidade (existem seis campos que podem ser preenchidos, definidos pelo usuário – peso, volume, medidas, valor, etc.); zona (especificada no cadastro de parâmetros); prioridade de atendimento (vai de A: mais baixa a Z: mais alta); tipo de operação (entrega ou coleta).

CADASTRO DE VEÍCULOS. Campos: nome; número (código ou placa); equipamento especial; capacidades (6 campos – definidos pelo usuário: peso,

volume, medidas, valor máximo transportado, etc.); custo (por km, por hora rodada, hora extra, espera, etc.); horário de início; horário de término; tempo de descanso; tempo que pode ser dirigido continuamente; se pode ficar mais de um dia trabalhando; tempo de despacho; tempo de redespacho; tempo mínimo de despacho; tempo máximo de despacho; tempo mínimo de redespacho; tempo máximo de redespacho; de onde sai (coordenadas x,y / depósito de origem); para onde vai (coordenadas x,y / depósito de destino final).

PARÂMETROS. Campos: escala do mapa (km/unid.); horas trabalhadas por dia; zona (tipo; velocidade; limitante).

CADASTRO DE DISTÂNCIAS. Cadastro de distâncias reais entre cidades; se não houver nada cadastrado o sistema se encarrega de realizar o cálculo da distância.

Saídas

O relatório final com os dados do roteamento pode ser definido pelo usuário com os dados que ele desejar (veículo; quilometragem total percorrida pelo veículo; nome do cliente; quilometragem parcial; CGC; volume; peso; valor; horário de chega; horário de saída; etc.; ou seja, qualquer dado cadastrado ou que pode ser dado como resposta pelo sistema pode fazer parte do relatório). O formato do relatório é definido pelo usuário através de um editor (arquivo .rrp). Segundo o funcionário, é fácil formatar e exportar o relatório.

O usuário tem a opção de edição da rota após o processamento.

Benefícios alcançados com a utilização do sistema

Os benefícios alcançados com a utilização do sistema são: maior controle; manifestos gerados automaticamente pelo roteirizador com a ordem das entregas; agilidade na operação de entrega (pois o veículo é carregado na ordem das entregas); eliminação de erros; etc., conforme resumidos no quadro 3.

Quadro 3: Benefícios alcançados pela empresa com a utilização do *software TruckStops*

TAREFAS	ANTES	DEPOIS
Entregas efetuadas em 24h	92%	98%
Retorno de cargas	0,84%	0,02%
Tempo de carregamento	28 minutos	21 minutos
Liberação de documento	16 minutos	6 minutos
Saída dos veículos (mesmo horário de início de preparação dos veículos)	8:11 h	6:52 h

Em São Paulo, houve uma redução de 16,22% dos custos de entregas e um aumento de 11% do número de entregas realizadas por veículo.

Comentários

A empresa realiza a roteirização sem a utilização de uma base cartográfica em decorrência de dois aspectos:

- 1) Dificuldade de criação de mapas viários através do editor de mapas, devido à abrangência da área coberta no processo de roteirização (cidades do país inteiro, incluindo o nível interurbano e urbano);
- 2) Inviabilidade de implementação de uma base cartográfica no formato MapInfo sob o ponto de vista custo/benefício, visto que sua utilização não interfere no cálculo das distâncias entre os clientes que é feito por distância euclidiana. É conveniente dizer que isto pode representar uma estimativa irreal da matriz distância/tempo real para o algoritmo.

O procedimento adotado pela empresa consiste em cadastrar os pontos (filiais, pontos de coleta e pontos de entrega) utilizando suas coordenadas (fornecidas em mapas tradicionais). São atribuídas, através do cadastro de distâncias, as distâncias reais entre os pontos “fixos” no sistema (filiais). A roteirização é dividida em etapas:

- Só coleta: geralmente consiste em cargas unitárias, que depois são fracionadas no centro de distribuição. Além disso, os veículos que realizam as coletas são diferenciados daqueles que fazem as entregas;

- Transferência entre filiais: o maior ganho se dá nesta fase, onde a matriz distância/tempo fornecida ao algoritmo pode ser estimada de forma bem precisa, visto que estão cadastradas as distâncias reais;
- Só entregas: nesta fase, a localização dos clientes varia diariamente, no entanto, dado que a área de abrangência é de nível nacional, a roteirização baseada em distância euclidiana (através do cadastramento das coordenadas do cliente) é ajustada através de edições manuais por parte do despachante ou tomador de decisão.

6.1.2 Distribuidora de bebidas

Esta empresa possui cerca de 4500 clientes que devem ser visitados todos os dias por uma frota de 120 veículos, realizando uma distribuição mensal de 2.678.978 caixas.

Como se trata de uma distribuidora de bebidas, os clientes variam desde padarias e supermercados até pequenos bares e casas noturnas. Nota-se aqui variados horários de atendimento submetidos pelos clientes em função do seu horário de funcionamento. Todos os clientes devem ser visitados todos os dias, seja para executar as entregas ou para realizar o pedido. Estes clientes estão espalhados por toda a cidade de Ribeirão Preto/SP e cidades vizinhas. Portanto, a roteirização dos veículos se dá em nível urbano e interurbano.

Os veículos possuem quatro tipos de capacidade diferentes, sendo os clientes localizados na área de centro comercial atendidos pelo veículo de menor capacidade, em função de restrições de estacionamento e leis de trânsito para circulação de veículos.

Os produtos variam de acordo com o tipo de bebida (cerveja, vários tipos de refrigerante de vários tamanhos e chope) e de acordo com o tipo de embalagem (frascos descartáveis ou retornáveis). As coletas são realizadas no ato da entrega. O número máximo de clientes visitados por rota é 45, sendo que o veículo pode realizar

mais de uma rota por dia, e o motorista não pode ultrapassar uma jornada de trabalho de 8 horas. Para a empresa, o ganho se dá em função do melhor aproveitamento da frota, onde o volume é o que determina a utilização do veículo.

As informações e características que são relevantes para a roteirização são: um único depósito, demanda determinística, janela de tempo nos clientes, todos os clientes devem ser visitados, veículos de diferentes capacidades e sem divisão de compartimento, realizar entregas e coletas (mas não é *backhauling*), permitir mais de uma rota por veículo, considerar restrição veículo-endereço (no caso de estabelecimentos localizados nos centros comerciais), limites iguais de duração da rota (45 clientes por veículo).

Software utilizado pela empresa: RoadShow

O *RoadShow* é comercializado no Brasil pela Routing Systems Informática há 8 anos, e é o sistema para roteirização utilizado por empresas como a Coca-cola, Perdigão, Brahma, Skol, Souza Cruz, Grupo Abril, etc.

Os pontos avaliados no *software* foram:

- *Requisitos mínimos de hardware para instalação*: o sistema opera com micro IBM-PC Pentium 166, 64 Mb de memória RAM, monitor 17 polegadas ou acima
- *Ambiente de instalação*: sistema operacional MS-DOS; existe também a versão para Windows
- *Tipo de instalação*: local (*stand alone*)
- *Custo da versão*: US\$ 25 mil (incluindo custo de atualização da versão)
- *Tempo para implantação*: 3 meses
- *Tempo em operação*: 2 anos
- *Suporte operacional*: existem 2 versões do *software* na empresa (uma em DOS e outra em Windows) e 3 funcionários envolvidos para suporte operacional (2 turnos de trabalho)

- *Versão para avaliação antes da aquisição do software:* não foi disponibilizada uma versão “Demo”. A divulgação do produto é feita através da apresentação e demonstração do *software* pela empresa *Routing Systems Informática*
- *Decisão de compra:* padronização do sistema de informação do grupo ao qual a empresa pertence
- *Consultoria e Treinamento:* é dado um treinamento de 4 dias
- *Manual:* em português

Base cartográfica

O *software* não traz uma base cartográfica. Caso o usuário faça a opção pelo seu uso ele deve arcar com os custos adicionais (cerca de U\$300 para cada mapa de cidade) e na versão para Windows o sistema possui interface com o *software* MapInfo. Assim como no *TruckStops*, o mapa consiste apenas em um pano de fundo para facilitar a visualização das rotas, ou seja, o *software* não reconhece os mapas como uma rede (onde o usuário enxerga o mapa o sistema enxerga uma tela negra). No entanto, é possível construir (digitalizar) uma rede sobre o mapa (onde este é utilizado como referencial) atribuindo a cada *link* informações com relação à velocidade, mão de direção, etc. Desta forma, o usuário deve construir a rede e o mapa consiste apenas em um referencial para a sua digitalização.

Na versão para DOS o processo de construção da rede é todo via teclado enquanto na versão para Windows, tudo é feito apenas com o uso do *mouse*.

Inserção dos Pontos de Vendas (PV's)

Os pontos de vendas (clientes) são inseridos diretamente sobre a base cartográfica, após a construção da rede, com o auxílio do *mouse*. Na tela há um campo denominado “*ponto de venda*”/*inserir*, com o qual o *mouse* fica ativo para clicar sobre a base cartográfica, identificando o local onde o ponto de venda está situado. O próximo passo é ligar este ponto à rede, o que também é feito com o auxílio do *mouse*, clicando sobre o primeiro nó da rede mais próximo do ponto de

venda, depois no ponto de venda e depois no segundo nó mais próximo, atribuindo em seguida o sentido aos segmentos que resultaram (caso haja mão única de direção). O resultado deste procedimento pode ser visualizado na figura 6.

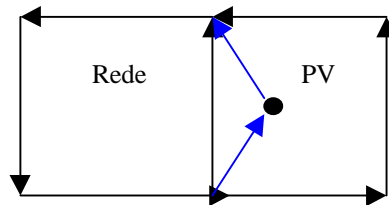


FIGURA 6 - INSERÇÃO DO PV NO *SOFTWARE ROADSHOW*

Vale dizer aqui que o sistema não aceita a inserção de PV's com distâncias menores que 50 metros entre eles. Assim, quando há mais de um cliente no mesmo quarteirão, eles são inseridos no mesmo ponto (ou seja, dentro de um PV podem existir vários clientes, sendo no máximo dez clientes na versão em DOS e infinitos na versão para WINDOWS).

A principal dificuldade consiste em localizar na base o ponto onde vai ser inserido o PV. Quando não há nenhum PV cadastrado, não há nenhum recurso de localização. O usuário tem que procurar a rua através de zoom. Quando há algum PV cadastrado, o usuário pode utilizá-lo como referencial para identificar (de acordo com a proximidade) os locais na base.

Ao cadastro do PV é atribuído automaticamente um número (número do PV) que se dá em função do seu ponto de localização (coordenadas). Neste mesmo cadastro podem ser inseridos o horário de atendimento (janela de tempo – o sistema oferece a opção de JT *soft* ou *hard*), os clientes associados a este PV, o “território” a que ele pertence e a frequência das visitas.

O território corresponde a uma função que agrupa os clientes em função de sua localização e/ou características em comum. O procedimento adotado pela empresa foi de agrupar os clientes em função de sua distribuição geográfica e horário de atendimento. Este procedimento foi feito manualmente pelo tomador de decisão,

baseado no seu conhecimento e intuição do modelo de otimização. Assim, cada conjunto de clientes ou pontos de vendas está associado a um território.

Por exemplo: o território 1 consiste no agrupamento dos pontos de vendas localizados no centro de Ribeirão Preto e que só podem ser atendidos até as 10 horas em função do horário permitido para estacionamento para carga e descarga.

Na figura 7, tem-se uma visão aproximada da tela para cadastro na versão para *Windows*.

FIGURA 7 - CADASTRO DOS CLIENTES NO PV NO *SOFTWARE ROADSHOW*.

Roteirização

Finalizado o processo de construção da rede, inseridos os pontos de vendas e realizada sua ligação à rede, é feito a exportação (*upload*) dos dados do cadastro de clientes gerando um arquivo tipo texto. Neste arquivo, um pedido simbólico de uma unidade é associado a cada cliente, para que durante o processo de roteirização todos os clientes fiquem “ativos”, ou melhor, façam parte do roteamento. Este procedimento é feito apenas como uma estratégia para agilizar o processo, visto que são cerca de 4500 clientes, o que tornaria inviável abrir o cadastro de cada cliente para colocar o seu pedido. Depois este arquivo é importado (*download*) para o *RoadShow*.

O próximo passo é realizar a “rolagem dos pedidos”. Este passo consiste em associar pedidos/ponto de venda/território. Todas estas funções estão disponíveis em campos na barra de ferramentas. A cada passo do processo avançado, os campos vão mudando de acordo com o procedimento que deve ser seguido. Depois da rolagem dos pedidos, o roteamento é processado por território.

Como pode ser notado, este sistema não utiliza veículos, e portanto não há restrição de capacidade. Para cada território vai ser gerada uma rota única e caso o usuário não faça um agrupamento dos clientes na forma de território, seria realizada uma única rota com todos os pedidos. Ao que tudo indica, o algoritmo consiste em uma heurística para o problema do caixeiro viajante.

Finalmente, os resultados são gerados na forma de relatório (seqüência, distância entre pontos de vendas, horário de atendimento e quilometragem total) e a rota pode ser visualizada sobre o mapa. O usuário tem a opção de edição manual da rota. Vale dizer que não é utilizada janela de tempo para realização do roteamento pela empresa, embora o sistema forneça esta opção.

O tempo de processamento é satisfatório. Para se ter uma idéia, os 4483 clientes estão divididos em 12 territórios, e para o maior território (que abrange uma área de quase metade da cidade de Ribeirão Preto), o processamento leva cerca de 5 minutos.

Na verdade, o *RoadShow* é utilizado por esta empresa apenas para gerar resultados com relação ao seqüenciamento de atendimento dos pedidos. A empresa possui um sistema integrado chamado RISAS BASE que faz controle de estoque, controle financeiro e roteirização. A roteirização é realizada com o auxílio dos *software RoadShow* e *Dispatching*. Enquanto o *RoadShow* é responsável pelo seqüenciamento das visitas, o *Dispatching* vai alocar os veículos para o atendimento da demanda. Assim, somente no *Dispatching* é que vai ser levada em consideração a quantidade real dos pedidos para os respectivos clientes que devem ser atendidos

naquele dia específico e a capacidade do veículo que vai ser alocado para realizar a entrega.

A roteirização é realizada no *RoadShow* somente quando há algum cliente novo. O arquivo com a seqüência das entregas é exportado para o *Dispatching*, e um arquivo previamente preparado pelo Departamento de Informática com os pedidos dos clientes também é utilizado pelo *Dispatching*. Para cada território existe uma lista de veículos com ordem de prioridade para atender às entregas. O *Dispatching* utiliza todas estas informações para realizar a roteirização, utilizando a seqüência fornecida pelo *RoadShow* (ignorando os clientes caso não haja entrega ou visita a ser realizada) até que a capacidade do veículo seja atingida. Esta capacidade foi definida como no máximo 45 visitas ou então, até atingir a cubagem do caminhão em função da quantidade de caixas (foi definido um fator de cubagem para a caixa). Quando um veículo atinge a capacidade máxima, o outro veículo da lista de prioridades começará a realizar a entrega a partir do ponto onde o último veículo parou, assim como demonstrado na figura 8.

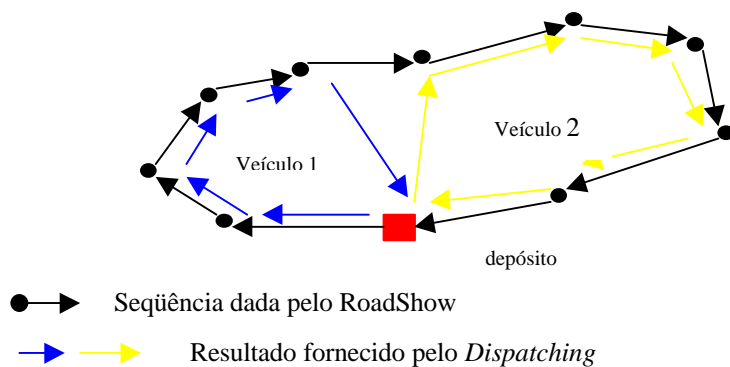


FIGURA 8 - EXEMPLO DE CONSTRUÇÃO DE ROTAS EM ROTEIRIZAÇÃO POR TERRITÓRIO NO SOFTWARE ROADSHOW.

Resumindo, a roteirização de veículos na empresa é realizada da seguinte maneira:

- 1) O vendedor possui um dispositivo eletrônico – o “*palm top*” - onde durante a visita aos clientes é inserido o seu pedido e a data de entrega, que geralmente ocorre no dia seguinte.

- 2) No final do expediente, o *palm top* é deixado no Departamento de Informática (DI), onde são descarregadas as informações no computador e é preparado um arquivo tipo texto com o código do cliente e o seu pedido.
- 3) Caso haja algum “novo cliente” que não esteja cadastrado no *RoadShow*, o DI manda para o Departamento de Logística um formulário preenchido (em papel) para que este possa ser inserido na base de dados do *RoadShow*. Depois de cadastrados os novos clientes (caso haja), é realizado o seqüenciamento das entregas com o auxílio do *RoadShow*.
- 4) Este seqüenciamento é exportado na forma de um arquivo tipo texto, que vai servir como dado de entrada no *Dispatching*, com o qual são alocados os veículos. Outro dado de entrada consiste no arquivo preparado pelo DI, com os clientes e seus respectivos pedidos.

Ganhos com o uso do Sistema

A análise dos benefícios trazidos pelo uso do sistema para a empresa foi realizada para um único território. Antes de utilizar o *RoadShow* eram necessários 26,5 veículos/dia para efetuar as entregas pertencente a esta rota e a ocupação média dos veículos era de 410 caixas. Com a utilização dos sistemas para roteirização (*RoadShow* e *Dispatching*) a frota necessária foi reduzida para 23 veículos e a ocupação média dos veículos oscila em torno de 480 à 515 caixas.

Em termos de quilometragem percorrida, a empresa não tem nenhum estudo realizado, mesmo porque não há dados para fazê-lo, uma vez que ela não faz este controle.

6.1.3 Distribuidora de produtos alimentícios

A empresa optou por não dar informações com relação a sua atividade operacional de distribuição. No entanto, foi demonstrado o *software* por ela utilizado para dar suporte ao planejamento diário da distribuição física dos produtos.

Software utilizado pela empresa: Trucks

Os pontos avaliados no *software* foram:

- *Empresa que desenvolveu:* Manugistics, Inc. (EUA)
- *Empresa que vende no Brasil:* Modus Logística Aplicada (SP)
- *Hardware instalado:* Pentium 2; 266 Mz; 32Mb; 4Gb
- *Tipo de instalação:* local (*stand alone*)
- *Ambiente de instalação:* DOS
- *Custo da versão:* mais de US\$ 50 mil (custo da versão básica)
- *Módulo:* pode ser inserido um módulo de gerenciamento de recursos para alocação dos motoristas (programação de motoristas), mas a empresa não possui este módulo
- *Tempo para implantação:* levou 3 anos para implantar toda a rede
- *Tempo em operação:* 5 anos
- *Tempo necessário para treinamento básico:* de 6 meses a 1 ano (treinamento completo); funcionários mais recentes levaram 3 meses para ler o manual e tiveram um curso de 1 semana (na Modus Logística Aplicada) para aprender a manipular a ferramenta
- *Suporte operacional:* existem 2 funcionários envolvidos e 3 versões do *software*. A primeira versão consiste nesta que está sendo descrita – a mais antiga. A segunda, consiste de uma versão, também em ambiente DOS, um pouco mais “evoluída”, mas que durante os testes apresentou problemas de duplicação de pedidos. Como isto não foi solucionado, tiveram que abandonar o seu uso; a terceira versão acabou de ser adquirida, consiste em uma versão para *Windows NT*, e está em fase de testes
- *Tempo necessário para utilização efetiva de toda a capacidade do software (do ponto de vista do usuário):* de 6 meses a 1 ano
- *Manual:* o manual é em inglês, mas existe uma tradução em português. É um manual bem básico (segundo o funcionário) e traz apenas algumas telas como exemplos explicativos
- *Decisão de compra:* disponibilidade no mercado

Interface Gráfica

- versão *Windows NT*: formato *MapInfo*
- versão *DOS*: digitalização das malhas viárias utilizadas (que podem ser tanto rodoviárias quanto urbanas). A cada nó são atribuídos: as coordenadas (latitude e longitude); o nome da cidade em que ele está localizado; UF (Unidade da Federação); e também os atributos dos segmentos que forma com os nós vizinhos. A tela é composta como mostrado na figura 9.

De	Nó	___	Cidade	_____	UF	___	Latitude	_____
							Longitude	_____
Para			distância		tipo		direção	nome
	Nó	__	_____		___		_____	_____
	Nó	__	_____		___		_____	_____
	...							

FIGURA 9 - TELA DO CADASTRO PARA CONSTRUÇÃO DA BASE CARTOGRÁFICA NO *SOFTWARE TRUCKS*.

Quando o campo distância é preenchido, o cálculo será feito levando em consideração este valor cadastrado, caso contrário, o cálculo será automático, utilizando as coordenadas e distâncias euclidianas. O campo “tipo” diz respeito à velocidade, que é cadastrada como na figura 10.

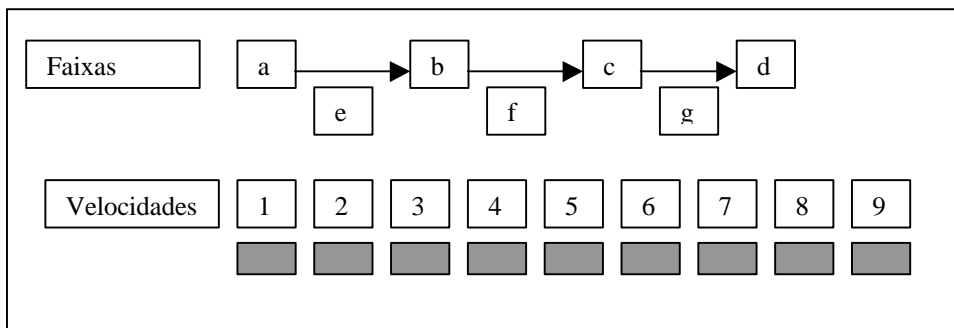


FIGURA 10 - CADASTRO DAS FAIXAS DE VELOCIDADES EM FUNÇÃO DA DISTÂNCIA PERCORRIDA, NO *SOFTWARE TRUCKS*.

Nos quadrados de cor cinza podem ser cadastrados nove “tipos” (valores) de velocidade, sempre com dois dígitos.

O campo “faixas” corresponde a diferenciação da velocidade em função da faixa de distância que será percorrida no *link* durante a roteirização. Os campos “a”, “b”, “c” e “d” devem ser preenchidos com qualquer número entre 0 e 9999 (que correspondem à distância), e os campos “e”, “f” e “g” devem ser preenchidos com um número entre 1 e 9 (que corresponde ao tipo de velocidade). Assim, por exemplo, se forem inseridos os valores 0, 5, 50, nada, 1, 3 e 5 para os campos “a”, “b”, “c”, “d”, “e”, “f” e “g”, respectivamente, tem-se que, se for percorrida uma distância entre dois pontos de parada menor ou igual a 5 Km, a velocidade será do tipo 1; se for percorrida uma distância entre 5 e 50 Km, a velocidade será do tipo 3; e se for percorrida uma distância maior que 50 Km a velocidade será do tipo 5.

Parametrização da rede

- Minimizar custo / distância / tempo (C/D/T): (___)
- Custo por milha: _____
- Custo por hora: _____
- Caminho através do nó (Y/N): (___) [este campo define se vai passar no nó da rede mais próximo ao cliente (Yes), ou se vai efetivamente até o nó onde o cliente está cadastrado (No)]
- Considerar um desvio da linha reta: (___%) [é uma forma de inserir uma porcentagem a mais na quilometragem calculada, para computar os desvios impostos pela rede em relação à distância euclidiana]

Entidades cadastradas

As entidades levadas em consideração neste sistema são: praças; subpraças; depósitos e veículos. Os clientes estão agrupados em praças, que nada mais são do que cidades. Quando a cidade é maior, como São Paulo, Campinas e Ribeirão, os clientes são agrupados em subpraças, que são bairros. Assim, o detalhamento maior do seqüenciamento das entregas diz respeito aos bairros que devem ser visitados.

“Location” ou Praças

- Cidade, Unidade da Federação
- Código
- Latitude
- Longitude
- Caminhões disponíveis (quando este campo não é preenchido com o código do caminhão que é compatível com a cidade, ele entende que qualquer caminhão possa fazer a entrega)
- Razão de entrega (pedidos atendidos/hora)
- Zona de entrega (região – podem ser cadastrados 5 tipos de zonas para cada praça, de modo que a praça possa pertencer a diferentes regiões para uma possível simulação)
- Horário de atendimento
- Pode ser o primeiro local a ser visitado
- Pode ser o último local a ser visitado
- Seqüência

Subpraças

- possui os mesmos campos de praça, mais o campo bairro

Depósitos

- Cidade, UF
- Código
- Latitude
- Longitude
- Caminhões disponíveis
- Zona de entrega
- Horário de funcionamento
- Capacidade de despacho (peso/h)
- Máximo de tempo que o motorista dirige

- Quilometragem máxima que o motorista dirige
- Horário de pernoite
- Máximo de espera por parada
- Distância máxima da rota
- Distância mínima da rota
- Tempo máximo na parada
- Tempo mínimo na parada
- Custo da hora do motorista
- Custo de hora extra
- Outros custos que não são utilizados pela empresa

Veículos

- Tipo de veículo
- Código do veículo
- Praça(s) ou subpraças que ele atende (o caminhão pode ser associado à praças ou subpraças, em decorrência do tipo e tamanho do veículo e característica do local, por exemplo, as peruas são associadas às subpraças centrais das grandes cidades devido ao problema de estacionamento)
- Capacidade (peso e volume)
- Número máximo de entregas
- Quantidade de veículos deste tipo
- Disponibilidade

Importação de dados

O arquivo de importação é do tipo texto e possui um formato padrão. Ele possui as seguintes informações com relação ao pedido:

- código da praça ou subpraça
- origem
- destino
- número de entregas

- peso
- volume
- valor
- seqüência
- tipo de operação

Parametrização da roteirização

- Minimizar distância/custo/tempo (D/C/T): (___)
- Considerar as estradas em uma única mão de direção (Y/N): (___)
- Velocidades em horário de *Rush*: (___%) atribui uma penalidade na velocidade em função dos horários acima ou abaixo da JT
- Classe da zona
- Violação da JT (Y/N): (___)
- Definir rota esqueleto (Y/N): (___)
- Deletar paradas sem pedidos (Y/N): (___)
- Seqüenciamento pré definido dos pedidos
- Considerar pedidos não roteirizados (Y/N): (___)
- Quantidade de interações
- Máximo de pedidos por rota
- Máximo de rotas geradas
- Máximo de paradas
- Tempo mínimo de viagem entre paradas
- Quantidade de pedidos que se deseja na ida e na volta: 0-100%
- Parâmetro da rota (relacionado com a configuração geométrica da rota): 0-100%
- Tempo de serviços entre janelas
- Janela de custo (*drive time/wait*): (___)
- Janela de tempo (rígida ou flexível): (___)
- “Quebrar” a rota em função da distância (*splitting*)

Saídas

As saídas geradas pelo sistema consistem em um relatório e o mapa com as rotas desenhadas sobre a malha viária. O relatório contém as seguintes informações:

- Veículo: Seqüência das praças ou sub-praças; peso; valor; número de entregas por praça ou sub-praça; horário de chegada; horário de saída; custo de utilização do veículo; custo do motorista.

O usuário tem a opção de edição manual da solução. O relatório pode ser exportado, através de um arquivo tipo texto gerado com um formato padrão.

Ganhos com o uso do sistema

Não foi realizado nenhum estudo pela empresa sobre o retorno financeiro com o uso do sistema. No entanto, é assegurado que o sistema já se pagou a muito tempo pois, onde antes havia cinco funcionários trabalhando, hoje existem apenas dois e trabalham com muito mais confiabilidade e controle.

Comentários

Como pode ser notado, a quantidade de parâmetros que devem ser ajustados é muito grande. Uma simples (e pequena) mudança em um destes parâmetros altera muito a solução apresentada pelo sistema. É um sistema difícil de ser calibrado. Mesmo a empresa que é responsável pela venda do produto aqui no Brasil não tem conhecimento de que forma muitos destes parâmetros interferem na qualidade da solução e como elas devem ser ajustados.

6.1.4 Conclusão

As empresas em questão realizam a roteirização levando em consideração janela de tempo ou restrição de comprimento da rota, consideram tempo de serviço nos clientes, utilizam veículos de diferentes capacidades, consideram uma rede viária para os cálculos de roteamento, entre outras coisas. No entanto, nenhum dos sistemas

por elas utilizados leva em consideração todas estas características simultaneamente, de forma adequada, fato que pode ser notado na maneira como elas operam seus sistemas, na prática.

No caso da transportadora, ela opera o sistema adotando o cálculo por distância euclidiana. No caso da distribuidora de bebidas ela deve pré-agrupar seus clientes em “territórios” como função de sua dispersão geográfica e janela de tempo, formulando o problema teórico como um simples problema do caixeiro viajante com restrição de comprimento de rota (limitada pelo atendimento de no máximo 45 clientes por rota), e a capacidade do veículo não é considerada.

Percebe-se que as ferramentas utilizadas nem sempre são adequadas as situações para as quais elas são aplicadas. É comum as empresas adaptarem seus problemas aos sistemas, ao invés do sistema ser apropriado, de forma a levar em consideração as características relevantes para sua solução.

6.2 TRANSCAD VERSÃO 3.2

O *software TransCAD* consiste em um Sistema de Informação Geográfica, aplicado à área de transportes, que incorpora, além das funções básicas de um SIG, rotinas específicas para soluções de problemas de logística, de pesquisa operacional e transporte em geral.

Este *software* possui um conjunto de ferramentas bastante robusto que resolvem vários tipos de problemas de roteirização de veículos. Estas ferramentas são usadas para preparar os dados, resolver o problema e gerar os resultados das rotas e programação dos veículos, tanto na forma de relatório quanto na forma gráfica.

As características que podem ser levadas em consideração, pelo *TransCAD*, no procedimento para resolução do problema de roteirização de veículos, são:

- 1) *Múltiplos depósitos*: é possível determinar antes da roteirização quais as paradas que serão atendidas por um determinado depósito; ou deixar que o próprio sistema se encarregue de alocar as paradas ao depósito mais adequado.
- 2) *Janela de tempo rígida*: é definida para todas as paradas em função de restrições de horários de atendimento. É atribuída ao depósito também, em função do seu horário de funcionamento, ou em função da jornada de trabalho do motorista.
- 3) *Tempo fixo de serviço*: corresponde ao montante de tempo requerido em cada parada, independente da quantidade de produto (ou serviço) demandada. É considerado, por exemplo, como um tempo de espera em filas para descarregar o veículo, ou o tempo para colocar o veículo em uma doca de descarga e verificar a mercadoria.
- 4) *Tempo por unidade*: tempo necessário para descarregar (ou carregar) cada unidade da mercadoria demandada.
- 5) *Restrição de comprimento total da rota*: esta restrição é dada em função do tempo máximo permitido para realizar uma rota.
- 6) *Frota heterogênea de veículos*: pode considerar veículos de diferentes capacidades.
- 7) *Paradas “backhaul”*: é possível realizar coletas em algumas paradas depois de realizar todas as entregas.

Existem algumas variações do problema que não podem ser resolvidos pelo *TransCAD*. Alguns exemplos são:

- 1) *Mistura de produto*: quando diferentes produtos devem ser entregues com o mesmo veículo, mas alguns veículos possuem restrição de tipo de produto que podem carregar.
- 2) *Início - término da rota*: veículos que não precisam retornar ao depósito no final do roteamento, ou podem visitar os depósitos durante o roteamento.

- 3) *Outras restrições*: é possível predeterminar uma ordem exata da visita para certas paradas.

O procedimento para resolver um problema de roteirização utilizando o *TransCAD*, consiste basicamente em quatro etapas:

Preparação dos dados de entrada.

É necessário criar um arquivo geográfico que contenha as informações com relação à localização de cada depósito e paradas, seguidas de informações sobre a demanda, e outras.

O(s) depósito(s) e paradas devem estar no mesmo arquivo geográfico (camada de paradas), que deve conter determinados campos necessariamente especificados para a roteirização e outros campos que são opcionais, dependendo das características do problema. A camada de paradas deve ter os campos apresentados no quadro 4.

Quadro 4: Cadastro das paradas no *software TransCAD*.

Campo	Tipo	Conteúdo
ID	Inteiro	Um número exclusivo que identifica cada parada
Nome	Caracter ou inteiro	Um nome ou um número que é usado para identificar a parada no relatório da rota
Demanda	Real	A demanda em cada parada
Horário de abertura	Inteiro	O primeiro horário em que a parada pode ser servida
Horário de fechamento	Inteiro	O último horário em que a parada pode ser servida

Dependendo do problema, os campos do quadro 5 podem ser também necessários.

Quadro 5: Campos opcionais no cadastro de paradas do *software TransCAD*.

Campo	Tipo	Conteúdo
Node_ID	Inteiro	O ID do nó da rede mais próximo de cada parada
Tempo fixo	Inteiro ou real	O tempo mínimo requerido para servir cada parada (em minutos)
Tempo por Unidade	Inteiro ou real	O tempo de serviço requerido para cada unidade de demanda (em minutos)
Depósito atribuído	Inteiro	O ID do depósito que vai servir a parada

É necessário também preparar uma tabela de veículos, que deve conter as informações apresentadas no quadro 6.

Quadro 6: Cadastro de veículos no *software TransCAD*.

Campo	Tipo	Conteúdo
Depot_ID	Inteiro	O ID do nó onde o depósito está localizado para cada tipo de veículo
Tipo	Inteiro	Código do veículo que deve ser único para cada depósito
Capacidade	Real	A capacidade de cada tipo de veículo
Num_veic	Inteiro	O número de veículos de cada tipo
Custo	Real	O custo de operação/aquisição/aluguel de cada tipo de veículo

Criação da matriz para roteirização

A matriz de roteirização é a primeira entrada para o procedimento de roteirização de veículos, e contém informações da distância e tempo de viagem entre cada depósito e parada e entre todos os pares de paradas. Esta matriz pode ser calculada através de dois métodos: usando uma rede ou usando a conexão em linha reta entre dois pontos (métrica euclidiana).

Para a criação da matriz devem ser escolhidas as paradas e depósitos a serem incluídos, bem como o método utilizado para calcular a matriz. É possível incluir todas as paradas e todos os depósitos, ou criar uma seleção do conjunto de paradas e depósitos que serão incluídos.

Quando é usada uma rede para criar a matriz, o programa calcula as distâncias e tempos de viagem através do menor caminho entre cada par de paradas e entre as paradas e o depósito. Quando os pontos que representam as paradas e depósitos não estão diretamente sobre a rede, o cálculo é feito entre os nós da rede mais próximos a estes pontos. Assim, a rede deve conter informações de distância e tempo de viagem para cada segmento. É possível atribuir também mão de direção aos segmentos. Quanto mais detalhada for a rede, mais precisas serão as estimativas do tempo de viagem e distância.

Quando é utilizada a métrica euclidiana para calcular a distância entre os pontos, o tempo de viagem é estimado através de uma velocidade média inserida

pelo usuário. Rotas criadas através deste procedimento não são exibidas como um sistema de rotas sobre um mapa.

Resolução do problema de roteirização de veículos

Para resolver o problema de roteirização de veículos com janela de tempo, deve-se seguir os seguintes passos:

- 1) escolher a tabela de veículos;
- 2) escolher a camada que contém as paradas e depósitos e especificar:
 - o conjunto de paradas que devem ser visitadas (seleção ou todas) – caso a roteirização inclua *backhaul* (opcional), deve-se criar também uma seleção destas paradas;
 - o(s) depósito(s) - caso exista mais de um depósito, é possível incluir todos ou selecionar os que se desejar incluir. Se a seleção de depósito incluir mais de um depósito é possível atribuir as paradas que serão servidas pelos depósitos ou deixar que o próprio sistema o faça durante a resolução do problema;
 - a janela de tempo nas paradas e depósito(s): este campo deve ser especificado. Se não existe restrição de janela de tempo, são atribuídos horários de abertura e fechamento arbitrários, de forma que o intervalo de tempo permitido não restrinja a roteirização;
 - o tempo fixo de serviço (opcional);
 - o tempo de serviço por unidade demandada (opcional);
 - o número de horas no comprimento máximo da rota: este número é aplicado sobre todas as rotas. Caso esta restrição não seja desejável, atribui-se um número alto.

Como padrão, o *TransCAD* roteiriza primeiro os clientes que estão mais distantes do depósito, mas o método de inicialização pode ser mudado pelo usuário e a prioridade de inserção dos clientes nas rotas será dada em função do menor prazo final para o atendimento da parada.

Se a matriz de roteirização foi criada a partir de uma rede, o resultado da roteirização pode ser convertido em um sistema de rotas, o que permite mostrar as rotas sobre um mapa. O sistema permite editar e combinar estas rotas.

Os resultados gerados pelo *TransCAD* consistem em:

- Resumo informativo do método utilizado para calcular a matriz: baseado em rede ou linha reta entre pontos; minimizar distância ou tempo; número de depósitos, número de paradas.
- Resumo informativo das características levadas em consideração na roteirização: 1) entradas: nome do arquivo da matriz; camada de paradas (seleção das paradas e depósitos); demanda total; tabela de veículos; capacidade total dos veículos; restrição de duração da rota; método utilizado para iniciar o procedimento; 2) saídas: tempo total; tempo total de viagem; tempo total de espera; tempo total de serviço; tempo da rota mais longa; distância total percorrida; número de rotas; demanda total servida; número de paradas visitadas; porcentagem de utilização da frota.
- Itinerário com a descrição de cada rota: número da rota; tempo total da rota; tipo do veículo utilizado; distância total; capacidade do veículo; carregamento do veículo; seqüência e programação das visitas.
- Caso tenha sido utilizado uma rede, a rota é mostrada sobre o mapa.

6.3 DELIVERY PROFESSIONAL

Este *software* consiste em uma ferramenta específica para problemas de roteirização de veículos e algumas de suas variações, dando apoio à decisão de escolha da rotina operacional de coleta e/ou distribuição física. Suas características estão mostradas no quadro 7.

Quadro 7.: Características do *Software Delivery*

CARACTERÍSTICAS	ALTERNATIVAS
Recursos e características	Diferentes tipos de veículos Um único depósito Coletas e entregas, “backhauls” Janela de tempo Tempos de carga e descarga Tempo de espera em fila Data e hora máxima para entrega Velocidades variáveis por tipos de vias da base cartográfica (pista dupla, pista simples, pista de terra) Limites de capacidade (peso e volume) Horários de início da viagem
Informações do roteiro (restrições e condicionantes)	Jornadas de múltiplos dias Possibilidade de roteirização por zona (região) Considera distância aérea ou terrestre Possibilidade de violação da janela de tempo Considera jornada de trabalho do motorista Comprimento máximo da rota
Funções-objetivo	Minimizar distância Minimizar tempo de viagem
Resultados	Relatório de visitas Relatório de utilização do veículo Relatório com dados gerais sobre o roteamento (custo, quilometragem total, taxa de utilização da capacidade do veículo – peso e volume, outros) Roteiros gráficos Alteração manual da solução

Fonte: adaptado de Assad (1986).

O programa é composto por uma base cartográfica vetorizada de todas as cidades, rodovias e hidrovias do Brasil, provida de recursos de localização dos dados nela cadastrados. Esta base não pode ser editada pelo usuário.

Para a construção das rodovias da base cartográfica, foram utilizados mapas fornecidos pelo DER/Ministério dos Transportes, de 1993, e mapas rodoviários dos estados, fornecidos pela AB-DER (Associação Brasileira dos Departamentos Estaduais de Estradas de Rodagem): Acre - 1992; Amapá - 1991; Amazonas – 1991; Bahia – 1994; Ceará – 1992; Distrito Federal –1994; Espírito Santo – 1993; Goiás – 1995; Mato Grosso – 1990; Mato Grosso do Sul – 1994; Minas Gerais – 1995; Pará – 1987; Paraíba, 1991; Paraná – 1993; Pernambuco – 1994; Piauí – 1992; Rio de Janeiro – 1992; Rio Grande do Norte – 1993; Rio Grande do Sul – 1994; Rondônia – 1993; Roraima – 1995; Santa Catarina – 1990; São Paulo – 1996; Sergipe – 1994;

Tocantins - 1991. Na construção dos mapas de hidrografia foi utilizado o Mapa de Divisão Política do Brasil (IBGE, 1996).

Vale dizer que os dados cadastrados nesta base cartográfica incluem também informações com relação aos serviços disponíveis (como por exemplo, postos de gasolina, pedágios, postos policiais, aeroportos). Embora possam não conter todos os existentes, pelo menos os principais são possíveis de serem localizados. Estes dados foram obtidos por meio de pesquisa direta junto aos órgão competentes (IBGE, Correios, companhias telefônicas, entre outros).

Assim, a matriz distância que vai servir como entrada para o algoritmo será estimada através do menor caminho percorrido nas estradas, entre as cidades cadastradas como clientes. A matriz tempo será estimada levando em consideração a distância e as velocidades cadastrada pelo usuário, em função do veículo e tipo de estrada (pista dupla, simples e terra). O usuário pode ainda bloquear trechos de rodovias, caso queira excluí-los do roteamento. Este método de cálculo da matriz distância/tempo corresponde ao “processamento terrestre”. O *software* também tem a opção de “processamento aéreo”, onde a matriz distância é calculada pela métrica euclidiana entre dois pontos. A matriz tempo usa uma velocidade média padrão de 80 Km/h, mas pode ser mudada pelo usuário.

Entrada de dados

Os dados do roteamento são inseridos neste sistema através de 5 cadastros:

- 1) Cadastro de depósito – campos: código do depósito; nome; endereço (opcional); estado; cidade;
- 2) Cadastro de regiões – descrição da região e sobretaxas de fretes (caso haja);
- 3) Cadastro de clientes – campos: código de identificação, localização dos clientes através do: estado; cidade; bairro (opcional); região (opcional); endereço (opcional); telefone (opcional); prioridade de atendimento (baixa, média ou alta); tempo de espera em fila. Aqui também é inserida a

janela de tempo, no campo “agenda do cliente”. Caso exista mais de um cliente na mesma cidade, é possível cadastrar a distância entre eles ou definir uma seqüência de entrega, como forma de solucionar o problema de roteirização urbana.

É claramente perceptível o quanto esta base limita a utilização do *software*, visto que não é viável a sua utilização quando se trata de roteirização urbana ou de pontos que não aqueles que possam ser relacionados às cidades cadastradas. Para ser mais geral, é possível realizar uma roteirização qualquer, desde que exista uma matriz distância dos “pontos” que devem ser visitados e o depósito. Isto pode ser feito, cadastrando-se vários clientes (que vão corresponder aos pontos) em uma mesma cidade (qualquer uma, pois será fictícia) e inserir as distâncias entre eles (matriz distância). Considerando que seja uma roteirização de 200 pontos (o que parece razoável para uma empresa comprometida com a roteirização de veículos em nível urbano), isto representa $(200)*(200-1)/2 = 19.900$ números que deverão ser inseridos pelo usuário como matriz distância.

No entanto, quando se trata roteirização interurbana, com alguns clientes na mesma cidade, o sistema é bastante amigável e fácil de manipular. Pode ser fornecida ao sistema a seqüência das visitas aos clientes localizados na mesma cidade, baseado no conhecimento do motorista ou do tomador de decisão.

- 4) Cadastro de visitas – campos: código da visita; dados do cliente (que pode ser trazido do cadastro de cliente); local da visita (caso não seja na mesma localidade em que o cliente se encontra); e dados da visita: data/hora máxima da visita (opcional), tempo de operação (carga e descarga), prioridade, tipo de operação (coleta ou entrega), peso, volume e valor da mercadoria.
- 5) Cadastro do veículo – campos: código; placa; carga máxima; volume máximo; custo por quilômetro rodado; tipo de carroceria: rígida ou flexível (no caso de carroceria rígida e existência de coletas, o problema reconhecido pelo algoritmo consiste em “*backhaul*”, no caso de flexível,

as coletas podem ser realizadas antes das entregas); prioridade de uso; disponibilidade; velocidades médias (pista dupla, pista simples, pista de terra); capacidade do tanque de combustível; consumo médio (Km/l); tipo de combustível (diesel, álcool ou gasolina).

- 6) Jornada de trabalho do motorista – campos: máximo de horas trabalhadas; mínimo de horas trabalhadas; máximo de horas dirigidas; mínimo de horas dirigidas; máximo de hora extra; mínimo de hora extra.

Informações do Roteamento

Aqui é montado o roteamento de acordo com as características do problema. É onde são fornecidas as informações com relação às restrições e condicionantes dos roteiros. As possibilidades são as seguintes:

- 1) Calcular por distância aérea ou terrestre;
- 2) Violar ou não a janela de tempo;
- 3) Realizar o roteamento por região ou não. Quando o roteamento é realizado por região, o próprio sistema se encarrega de designar os caminhões para cada região, no entanto, é possível a intervenção do usuário, alterando manualmente a escolha;
- 4) Considerar ou não jornada de trabalho do motorista.

Saídas

- 1) Relatório de visitas – seqüência das visitas; horário de chegada; horário de saída; custo; quilometragem parcial percorrida; veículo que atendeu; data/hora máxima de entrega da mercadoria; tipo de violação, caso tenha ocorrido (por exemplo, violação do prazo de entrega).
- 2) Relatório de utilização do veículo – quilometragem total percorrida; tempo de viagem; custo da viagem; carga (peso e volume) transportada; horário de saída; horário de retorno.

- 3) Relatório com dados gerais sobre o roteamento (custo, quilometragem total, taxa de utilização da capacidade do veículo – peso e volume, outros).
- 4) Roteiros gráficos – mapa com a(s) rota(s) traçada(s). Alteração manual da solução – o roteamento pode ser alterado através da inserção de novas visitas; alteração da seqüência de atendimento; retirada de visitas.

6.4 RESUMO DAS PRINCIPAIS CARACTERÍSTICAS ENCONTRADAS EM TODOS OS *SOFTWARE* ANALISADOS NESTA SEÇÃO

A tabela 4 traz uma comparação entre os *software* descritos nesta seção, apontando suas características gerais.

Tabela 4: Software x Características gerais para roteamento

Software	Delivery	RoadShow	TruckStops	Trucks	TransCAD 3.2
Ambiente de instalação	WINDOWS	DOS WINDOWS	DOS WINDOWS	DOS WINDOWS	WINDOWS
Custo aproximado da versão	R\$ 3 mil	US\$ 25 mil	US\$ 17 mil	US\$ 50 mil	US\$ 11 mil
Distâncias e tempos de viagem	Distância euclidiana / Rede vetorizada; velocidade por tipo de veículo e padrão de rodovia (pista simples, pista dupla, pista de terra, etc.)	Coordenadas (distância euclidiana)/ rede construída	Coordenadas, zonas de velocidade e barreiras.	Coordenadas (distância euclidiana)/ rede	Coordenadas/ Rede
Janelas de tempo	Rígida	Rígida e Flexível	Rígida	Rígida e Flexível	Rígida
Múltiplas rotas por roteamento	Sim	Não	Sim	Sim	Sim
Múltiplos roteiros por veículo	Não	Não utiliza veículos	Sim	Sim	Não
Múltiplas bases	Não	Não	Sim	Sim	Sim
Permite Roteiros c/ pernoite	Sim	Sim	Sim	Sim	Sim
Entidades consideradas no roteamento	1) Depósito 2) Veículo 3) Cliente 4) Parada 5) Região - opcional	1) Depósito 2) Região - opcional 3) Cliente 4) Parada	1) Depósito 2) Veículo 3) Parada ou cliente	1) Praças 2) Subpraças 3) Depósitos 4) Veículos	1) Depósito 2) Parada ou cliente 3) Veículo
Coleta de retorno backhaul	Sim	Sim	Sim	Sim	Sim
Recurso gráfico	Sim	Sim	Sim	Sim	Sim
Mudanças manuais das soluções	Sim	Sim	Sim	Sim	Sim

O quadro 8 traz a classificação dos *software* que serão efetivamente aplicados neste estudo, segundo o esquema proposto por Desrochers *et al.* (1990):

Quadro 8: Caracterização dos *software Delivery* e *TransCAD* (Desrochers *et al.*, 1990).

CAMPO	SUBCAMPO	ELEMENTOS	<i>Delivery</i>	<i>TransCAD</i>
ENDEREÇO	Número de Depósito	1	Sim	Sim
		L	Não	Sim
	Tipo de Demanda	nó	Sim	Sim
		arco	Não	Não
		misto	Não	Não
		tarifa	Não	Não
		tudo entrega ou coleta	Sim	Sim
		"backhauls"	Sim	Sim
		determinística	Sim	Sim
	Restrição de Programação	estocástica	Não	Não
		sem restrição de horário	Sim	Não
		programação fixa	Sim	Sim
		janela de tempo simples	Sim	Sim
	Restrição de Seleção	janela de tempo múltipla	Sim	Não
		sem restrição (padrão)	Sim	Sim
subconjunto		Sim	Sim	
escolha		Não	Não	
VEÍCULOS	Número de Veículos	período	Não	Não
		c	Sim	Sim
		m (instância)	Sim	Sim
		nem todos veículos especificados devem ser utilizados	Sim	Sim
	Restrição de Capacidade	todos os veículos devem ser utilizados	Sim	Não
		sem restrição	Não	Não
		veículos de mesma capacidades	Sim	Sim
	Restrição de Mercadoria	veículos de capacidade diferentes	Sim	Sim
		sem divisão de compartimento	Sim	Sim
		multicompartimentos	Não	Não
	Restrição de Programação	especiais	Não	Não
		sem restrição de horário	Sim	Sim
		janela de tempo idêntica para todos os veículos	Não	Não
	restrição de duração das rotas	janela de tempo diferente	Não	Não
		sem restrição de tempo	Sim	Não
limites iguais		Sim	Sim	
CARACT. DO PROBLEMA	tipo de rede	limites diferentes	Não	Não
		com sentido	Não	Sim
		sem sentido	Sim	Sim
		mista	Não	Sim
		cálculo do tempo de viagem em função da distância calculada por métrica euclidiana	Sim	Sim

Quadro 8: Caracterização dos software *Delivery* e *TransCAD* (Desrochers *et al.*, 1990) (Continuação).

CAMPO	SUBCAMPO	ELEMENTOS	<i>Delivery</i>	<i>TransCAD</i>
CARACT. DO PROBLEMA	tipo de rede	cálculo do tempo de viagem em função da distância calculada sobre a rede.	Sim	Sim
	tipo de estratégia	sem quebra de demanda	Sim	Sim
		quebra da demanda no princípio	Não	Não
		quebra da demanda no final	Sim	Não
		sem restrição de backhails ou carga completa	Sim	Sim
		backhails (para roteirização em nó)	Sim	Sim
		carga completa (tarefa)	Não	Sim
		uma rota por veículo	Sim	Sim
		permite mais de uma rota/veículo	Não	Sim
		a rota começa e termina no mesmo depósito	Sim	Sim
	rotas múlti-depósito	Não	Sim	
	restrição endereço - endereço	sem restrição de precedência	Sim	Sim
		com restrição de precedência	Não	Não
		sem restrição depósito-endereço	-	Sim
		com restrição depósito-endereço	-	Sim
		sem restrição endereço-endereço	Sim	Sim
	com restrição endereço-endereço	Sim	Não	
	restrição endereço - veículo	sem restrição depósito-veículo	Não	Não
		com restrição depósito-veículo	Sim	Sim
sem restrição endereço-veículo		Sim	Sim	
com restrição endereço-veículo		Não	Sim	
restrição veículo - veículo	sem restrição veículo-veículo	Não	Sim	
	sincronização entre veículos (com restrição)	Não	Não	
OBJETIVOS	operador	minimizar a soma da função custo	Sim	Sim
		minimizar o valor máximo da função custo	Não	Não
	função	tempo de duração da rota	Sim	Sim
		custo do veículo	Sim	Sim
		limitação/penalidade do veículo	Sim	Não
		custo no endereço	Não	Não
		limitação/penalidade no ender.	Sim	Sim
	penalidades veículo	capacidade	Sim	Não
		janela de tempo	Não	Sim
		duração do percurso	Não	Sim
	penalidades endereço	janela de tempo simples	Sim	Sim
		janela de tempo múltipla	Sim	Não

7 ESCOLHA DAS CLASSES DE PROBLEMAS TESTADAS.

As características levantadas nos *software Delivery* e *TransCAD*, bem como as características mais críticas encontradas em problemas de roteirização reais (no sentido de dificultar a operação pelo “método intuitivo” do sistema de distribuição) foram decisivas para este processo de escolha.

O problema geral tratado é o Problema Clássico de Roteirização de Veículos (*um depósito, multi-veículos, problema de roteirização de nós*). Nestes problemas, clientes devem ser servidos por veículos de capacidade limitada a partir de um depósito central. O objetivo é encontrar um conjunto de rotas, onde cada rota começa e termina no mesmo depósito, depois de visitar um subconjunto de clientes. Os veículos possuem capacidade iguais, e cada cliente é visitado por um único veículo. A capacidade dos veículos não é excedida e a distância total das rotas deve ser a mínima possível. Cada cliente corresponderá a um ponto no plano, e a distância entre eles será dada pela distância euclidiana entre estes dois pontos. Também será considerado o problema formulado sobre uma rede viária, onde a distância entre os clientes corresponderá àquela fornecida pela base cartográfica. Variações deste problema considerando-se, além do aspecto espacial, aspectos temporais, também serão tratadas. Utilizando o esquema de classificação e a linguagem propostos por Desrochers *et al.* (1990), os conjuntos de problemas testes e as classes de problemas que ele representam são apresentadas nos itens a seguir.

7.1 OS QUATORZE PROBLEMAS TESTE DE CHRISTOFIDES, MINGOZZI E TOTH (1979)

Classificação do PRV: $1|c, \text{cap}| \nabla | \text{sum}T_i$;

$1|c, \text{cap}, \text{dur}| \nabla | \text{sum}T_i$.

Parece haver um forte consenso de que estes problemas testes são apropriados para este tipo de avaliação, pois foram explorados por muitos pesquisadores. O conjunto de quatorze problemas teste de Christofides, Mingozzi e Toth (1979) foi inicialmente utilizado pelos autores para testar duas novas heurísticas por eles propostas (*two-phase algorithm* e *incomplete tree search algorithm*). Os autores ainda aplicaram estes testes sobre o algoritmo de Clarke e Wright (1964), o de Mole e Jameson (1976) e sobre o algoritmo de varredura de Gillett e Miller (1974), comparando o desempenho das heurísticas através dos resultados encontrados com sua aplicação. Posteriormente, estes problemas testes foram aplicados nas novas heurísticas propostas por Fisher e Jaikumar (1981), Desrochers e Verhoog (1989), Altinkemer e Gavish (1991), Pureza e França (1991), Taillard (1992), Osman (1993), Gendreau, Hertz e Laporte (1994), Kelly e Xu (1999). Os valores das soluções encontradas por esses algoritmos encontram-se na Tabela 2.

Os problemas contêm entre 50 e 199 clientes, mais o depósito. Dentre estes 14 problemas, 7 são problemas básicos de roteirização de veículos ($1|c, \text{cap}| \nabla | \text{sum}T_i$). Os 7 problemas restantes usam as mesmas localizações dos 7 primeiros, mas possuem restrições de comprimento máximo das rotas ($1|c, \text{cap}, \text{dur}| \nabla | \text{sum}T_i$). Além disto, considera-se um tempo de serviço constante, cobrado por visitar cada cliente (que pode ser traduzido para uma unidade de distância δ). Assim, os problemas de 1 a 5, 11 e 12 possuem apenas restrição de capacidade. Os problemas de 6 a 10 são os mesmos de 1 a 5, no entanto possuem também restrição de comprimento máximo da rota e tempo de serviço; os problemas 13 e 14 são iguais aos problemas 11 e 12, mas nestes últimos há restrição de comprimento da rota e considera-se tempo de serviço nos clientes. Estas informações estão resumidas na tabela 5.

Tabela 5: Características dos problemas testes de Christofides *et al.*, 1979.

Número do Problema	Tamanho (n)	Capacidade do Veículo	Comprimento máximo da rota	Tempo de Serviço	Razão da Capacidade	Demanda total	Número de veículos
VRPNC1	50	160	∞	0	0,97	777	5
VRPNC2	75	140	∞	0	0,97	1364	10
VRPNC3	100	200	∞	0	0,91	1458	8
VRPNC4	150	200	∞	0	0,93	2235	12
VRPNC5	199	200	∞	0	0,98	3186	17
VRPNC6	50	160	200	10	0,80	777	6
VRPNC7	75	140	160	10	0,88	1364	11
VRPNC8	100	200	230	10	0,81	1458	9
VRPNC9	150	200	200	10	0,80	2235	14
VRPNC10	199	200	200	10	0,88	3186	18
VRPNC11	120	200	∞	0	0,98	1375	7
VRPNC12	100	200	∞	0	0,90	1810	10
VRPNC13	120	200	720	50	0,62	1375	11
VRPNC14	100	200	1040	90	0,82	1810	11

Nos problemas de 1 a 10, os clientes estão distribuídos uniformemente, enquanto nos problemas de 11 a 14 eles aparecem em agrupamentos. As coordenadas dos clientes foram geradas randomicamente, retiradas de uma distribuição uniforme entre $U[1,100]$, enquanto as coordenadas dos depósitos são escolhidas de $U[45,55]$. As demandas foram geradas em um intervalo $U[1,41]$ enquanto a capacidade dos veículos é fixa, obedecendo a uma razão ρ , que corresponde a razão entre a demanda total requerida e a capacidade total disponível, variando em $U[0,82; 0,97]$.

Os arquivos contendo estes testes estão disponíveis através da Internet, na OR-Library, e encontram-se no seguinte formato (tipo texto):

número de clientes; capacidade do veículo; tempo máximo de rota; tempo de serviço;
 depósito: coordenada x; coordenada y;
 para cada cliente: coordenada x; coordenada y; demanda.

Os clientes e depósito são considerados pontos em um plano e as distâncias entre eles são dadas pela distância euclidiana entre dois pontos.

7.2 CONJUNTO DE PROBLEMAS DE SOLOMON (1987)

Classificação do PRV: $1|jt|c, cap|\nabla|somT_i$

O desenvolvimento destes problemas testes destaca vários fatores que podem afetar o comportamento de heurísticas para roteirização e programação de veículos. Estes fatores incluem: dispersão dos dados geográficos, número de clientes servidos por um veículo, características das janelas de tempo, reproduzidas pela porcentagem e posicionamento de clientes com restrições de horários de atendimento, e o quão restritos são esses horários.

Foram gerados seis conjuntos de problemas: R1, R2, C1, C2, RC1 e RC2. Os dados usados para coordenadas e demanda dos clientes são baseados nos dados de alguns dos problemas do conjunto padrão de problemas testes para roteirização de Christofides, Mingozzi e Toth (1979). Os dados geográficos são randomicamente gerados: em uma distribuição uniforme (nos problemas correspondentes aos conjuntos R1 e R2); por agrupamento (problemas correspondentes aos conjuntos C1 e C2) e; semi-agrupamento (problemas correspondentes aos conjuntos RC1 e RC2). Este último conjunto contém uma mistura de dados gerados randomicamente em uma distribuição uniforme e por agrupamento.

O conjunto de problemas R1, C1 e RC1 têm um curto horizonte de tempo de programação. O comprimento da rota é limitado pela capacidade do veículo, permitindo que apenas alguns clientes sejam servidos pelo mesmo veículo. Em contrapartida, os conjuntos R2, C2 e RC2 têm um longo horizonte de tempo de programação; esta característica está vinculada aos veículos com grande capacidade.

Determinados os dados geográficos e de demanda, foram criados os problemas testes de PRPVJT (problemas de roteirização e programação de veículos com janela de tempo) através da geração de janelas com vários intervalos de tempo, para diferentes porcentagens de clientes. Em termos de densidade, isto é, a

porcentagem de clientes com janela de tempo, criou-se problemas com 25, 50, 75 e 100% dos clientes submetidos a esta restrição.

O método utilizado para gerar randomicamente as restrições de janelas de tempo, nos conjuntos de problemas R1, R2, RC1, RC2, foi: primeiro, selecionou-se a porcentagem de clientes a receberem janelas de tempo, digamos f , $0 < f \leq 1$. Depois, gerou-se n números randômicos, de uma distribuição uniforme sobre o intervalo (0,1) e sorteou-os. Desta maneira, criou-se uma permutação randômica de clientes, (i_1, i_2, \dots, i_n) , com i_j posicionado depois do j -ésimo número sorteado. Finalmente, atribuiu-se janelas de tempo aos clientes i_1, \dots, i_{n1} , com $n1$ eleito por ser o inteiro que mais se aproxima de $f*n$. A janela de tempo tem centro e intervalo gerados randomicamente. O centro da janela de tempo para o cliente i_j , para $j = 1, \dots, n1$, corresponde a um número gerado randomicamente de uma distribuição uniforme, dentro de um intervalo específico. Para criar o intervalo da janela de tempo, foi gerado randomicamente um número em uma distribuição uniforme para ser a metade do intervalo.

O método utilizado para gerar a janela de tempo, nos problemas C1 e C2, consistiu em executar a rotina do algoritmo 3-opt (Lin¹, 1965, *apud* Solomon, 1987) em cada agrupamento, para criar rotas e então programar horários, dando uma orientação para cada rota. As janelas de tempo foram geradas da seguinte forma: o centro corresponde ao horário de chegada em cada cliente; o intervalo e a densidade foram gerados da mesma maneira descrita anteriormente. O conjunto de problemas C1 e C2 é composto de casos bem estruturados, no sentido de que os clientes aparecem agrupados e suas janelas de tempo foram atribuídas em função do horário de chegada ao cliente, na solução do problema com o algoritmo 3-opt. Esta metodologia permite a identificação de soluções agrupamento-por-agrupamento muito boas, possivelmente ótimas, o que provê um meio adicional de avaliação da performance da heurística.

¹ LIN, S. (1965). Computer Solution of the Traveling Salesman Problem. *Bell System Tech. J.*, n. 44, 2245-2269.

Todos os problemas testes possuem 100 clientes que correspondem a pontos em um plano euclidiano. Os tempos de viagem entre os clientes são igualados às distâncias correspondentes. A frota de veículos assumida é homogênea.

A tabela 6 traz os resultados alcançados por algumas heurísticas encontradas na literatura, propostas para resolver os problemas de roteirização de veículos com janela de tempo.

Tabela 6: Comparação de heurísticas aplicadas sobre o conjunto de dados do Solomon.

Conjunto de problemas		PB	CR-as	TH	RT	TA	CR-TABU	RNETS
R1 12 problemas	No. de rotas	12,60	12,50	12,33	12,58	12,25	12,17	12,17
	Distância	1296,80	1308,82	1249,30	1197,42	1216,70	1204,19	1249,57
	Tempo computacional	11,32	2,86	25,25	45,00	229,56	149,85	44,28
R2 11 problemas	No. de rotas	3,00	2,91	3,09	3,09	3,00	2,73	2,82
	Distância	1117,70	1166,42	1007,30	954,36	995,38	986,32	1016,58
	Tempo computacional	39,44	3,93	23,15	163,33	337,20	169,88	6,64
C1 9 problemas	No. de rotas	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	Distância	838	909,8	830,8	828,45	828,45	828,38	830,06
	Tempo computacional	10,02	2,19	4,25	53,33	243,83	17,90	22,01
C2 8 problemas	No. de rotas	3,00	3,00	3,00	3,00	3,00	3,00	3,00
	Distância	589,90	684,10	654,20	590,32	590,30	591,42	591,03
	Tempo computacional	41,37	2,46	5,25	60,00	272,92	40,20	3,59
RC1 8 problemas	No. de rotas	12,10	12,38	12,00	12,38	11,88	11,88	11,88
	Distância	1446,20	1473,90	1438,00	1369,48	1367,51	1397,44	1412,87
	Tempo computacional	11,22	2,57	16,18	43,33	187,73	78,96	30,46
RC2 8 problemas	No. de rotas	3,40	3,38	3,37	3,62	3,38	3,25	3,25
	Distância	1360,60	1401,50	1243,70	1139,79	1165,62	1229,54	1204,87
	Tempo computacional	35,57	3,58	20,20	130,00	193,27	107,39	7,12

Fonte: Liu e Shen (1999)

PB=Potvin e Bengo (1996); CR-sa=Chiang e Russell (1996); TH=Thangiah *et al.* (1995); RT=Rochat e Taillard (1995); TA=Taillard *et al.* (1997); CR-tabu=Chiang e Russell (1997); RNETS=Liu e Shen (1999).

Como pode ser visto, esta tabela traz uma média dos resultados para cada conjunto de problemas (R1, R2, C1, C2, RC1 e RC2). As células destacadas representam as melhores soluções, para cada classe de problemas, encontradas na literatura revisada. Todos os artigos revisados apresentam a tabela de comparação

desta mesma forma, além de alguns resultados particulares encontrados para alguns problemas específicos. Por exemplo, em Liu e Shen (1999), encontram-se os resultados dos problemas RC202 e RC205. Em Russell (1995) estão apresentadas as soluções de R101, R102 e RC101.

No presente estudo, não são executados os 56 problemas testes, pois os *software* não resolvem problemas com longo horizonte de tempo, onde as janelas de tempo dos clientes são definidas em um intervalo que ultrapassa 24h. Sendo assim, foram executados apenas os grupos R1, C1 e RC1.

Os dados dos 56 problemas teste de Solomon pode ser encontrado em <http://www.cs.strath.ac.uk/~ps/GreenTrip/NewBest/SolProblems/>. Neste mesmo *site* podem ser encontrados os resultados de alguns destes problemas, conseguidos por um sistema integrado de suporte à decisão para logística em transporte, desenvolvido pela *Green Trip*.

7.3 CONJUNTO DE TESTE COM DADOS REAIS

O objetivo aqui é analisar a performance dos algoritmos quando é considerada uma rede viária no problema de roteirização de veículos. Para tanto, será aplicado um caso real de distribuição física de uma empresa situada em São Carlos-SP, utilizando a base cartográfica da malha viária desta cidade empregada em Pelizaro *et al.* (1998). Os resultados serão comparados com a quilometragem rodada pela empresa ao realizar suas entregas.

O problema a ser testado consiste no programa de distribuição de uma micro usina de beneficiamento de leite situada na zona rural do município. O objetivo é minimizar a distância percorrida, para diminuir custos de transporte e também, porque o leite e seus derivados são produtos de alta perecibilidade, exigindo o mínimo tempo de exposição à temperatura ambiente para garantir sua qualidade.

O problema aqui testado consiste na entrega diária de 2500 litros de leite e de 150 litros de iogurte, em 92 pontos comerciais na cidade (sendo estes bares, padarias, varejões, supermercados e sorveterias). A empresa possui sua própria frota de veículos, que corresponde a um caminhão isotérmico e uma perua. Com esta frota é possível atender a demanda realizando uma única viagem diária por veículo. A quantidade de produtos demandada por cada cliente varia a cada dia e depende do potencial de venda de cada ponto comercial, ou seja, a distribuição é pronta entrega, e a quantidade (já com uma previsão da demanda) é definida na hora da entrega. No entanto, sabe-se que a perua é responsável pelo atendimento de 22 estabelecimentos e o caminhão pelos outros 70 restantes.

Em alguns pontos comerciais há um intervalo de horário fixo para recebimento dos produtos distribuídos pela empresa, que deve ser respeitado a fim de satisfazer o cliente. Portanto, para o planejamento e otimização da rota serão levadas em consideração as janelas de tempo. Os veículos têm como ponto de partida um único depósito situado na cidade. É preferível que o trajeto mais longo seja realizado pelo caminhão, pois este é isotérmico e portanto conserva por mais tempo a temperatura dos produtos, além de ter maior capacidade de armazenamento de carga.

O tempo necessário para carga e descarga em cada ponto de distribuição é de aproximadamente dois minutos, e a velocidade média com que os veículos circulam é de 35 Km/h (tanto a perua quanto o caminhão).

A empresa já possui uma rota de distribuição que foi ajustada intuitivamente e experimentalmente para percorrer a menor distância, sem desrespeitar os horários estipulados pelos clientes para a entrega dos produtos. Seguindo esta rota, o tempo de ciclo para o caminhão é de 6 horas e 12 minutos, tendo início às 5h15m e término às 11h27m, percorrendo uma quilometragem de 66,78 Km. O tempo de ciclo para a perua é de 3 horas e 6 minutos, iniciando às 5h30m e terminando às 8h36m e percorrendo uma quilometragem de 22,96 Km.

O problema real encontrado pode ser formulado de duas maneiras. Uma maneira consiste em utilizar o procedimento adotado pela empresa, no qual são pré agrupados 22 pontos de entrega que devem ser efetuadas pela perua e não considerar a capacidade dos veículos (visto que esta é muito maior que a demanda requerida). Outra forma, é definir valores arbitrários para as capacidades dos veículos (22 e 70 - perua e caminhão, respectivamente) e quantidades de demandas (uma unidade para cada cliente, somando um total de 92), de forma que a perua fique responsável pelo atendimento de 22 estabelecimentos e o caminhão fique responsável por 70. Este último procedimento permite uma melhor análise do desempenho do algoritmo, uma vez que o próprio sistema se encarrega de definir os pontos que serão visitados por cada veículo.

Assim, as classes de problemas teóricos formuladas e analisadas com estes dados são:

- 1) $1, |j_t| = c, \text{ esp } |AV| \text{ som } T_i$ (com restrição veículo-endereço);
- 2) $1, |j_t| = c, \text{ cap}_i | \text{ som } T_i$ (sem restrição veículo-endereço, mas com restrição de capacidade dos veículos – frota heterogênea);

Estes casos também foram processados utilizando a métrica euclidiana para o cálculo das distâncias, o que classifica o problema como:

- 3) $1, |j_t| = c, \text{ esp } |AV| |\nabla| \text{ som } T_i$;
- 4) $1, |j_t| = c, \text{ cap}_i |\nabla| \text{ som } T_i$

Também foram desconsideradas as restrições de janela de tempo nos clientes, para todas as variações do problema, expandido a quantidade de classes testadas.

- 5) $1 | =, c, \text{ esp } |AV| \text{ som } T_i$;
- 6) $1 | =, c, \text{ cap}_i | \text{ som } T_i$
- 7) $1 | =, c, \text{ esp } |AV| |\nabla| \text{ som } T_i$;
- 8) $1 | =, c, \text{ cap}_i |\nabla| \text{ som } T_i$

No Apêndice encontra-se figuras que ilustram a disposição geográfica no plano dos pontos que representam os clientes e depósitos para todos os problemas teste que foram aplicadas nesta análise empírica.

7.4 OS TESTES COMPUTACIONAIS

Os dois *software* foram instalados em um microcomputador Pentium (r), tecnologia Intel MMX(TM), com 128 MB de memória RAM e 32 bits de memória virtual. O aplicativo da versão utilizada do *software Delivery* ocupa um espaço de 865 KB, enquanto a versão do *TransCAD*, 4630 KB. Os dois sistemas foram desenvolvidos em ambiente Windows.

7.4.1 Problemas Testes de Christofides, Mingozzi e Toth (1979)

As soluções apresentadas pelos *software TransCAD* e *Delivery* para este conjunto de problemas encontram-se nas tabelas 7 e 8, respectivamente. As tabelas trazem também as melhores soluções encontradas na literatura para este conjunto de problemas teste, resultado da aplicação de diversas heurísticas propostas para resolver o problema de roteirização de veículos.

Uma consideração que deve ser feita na análise da qualidade das soluções corresponde à técnica empregada na construção dos algoritmos comparados. A maioria dos *software* comerciais para roteirização emprega algoritmos baseados em métodos construtivos, em função dos tempos de processamento apresentados serem mais razoáveis. No entanto, as melhores soluções encontradas para estes problemas testes foram conseguidas através de algoritmos que empregam metaestratégias, tais como *simulated annealing* e busca tabu, os quais requerem tempo computacional maior que as heurísticas clássicas. Assim, melhores qualidades nas soluções são penalizadas por um maior tempo computacional.

Em função disso, além de avaliar a qualidade da solução apresentada pelos *software* com relação aos melhores resultados (o que representa uma válida medida de desempenho dos algoritmos testados), é justo que se faça também uma comparação com os resultados encontrados por uma “boa” heurística de construção, que é o caso da heurística proposta por Solomon (1987). As soluções encontradas para os problemas aqui analisados com a aplicação desta última heurística também estão mostrados nas tabelas 7 e 8.

Tabela 7: Soluções apresentadas pelo *software TransCAD* para os problemas propostos por Christofides *et al.* (1979).

Problema	Melhor solução	Solomon ¹ (1987)	TransCAD		
			Custo (km)	Número veículos	Tempo computacional (segundos)
VRPNC1	524,61	611,17	639,10	5	9
VRPNC2	835,26	996,44	985,70	10	24
VRPNC3	826,14	1048,02	1017,70	8	100
VRPNC4	1028,42	1297,72	1330,80	12	460
VRPNC5	1300,89	-	1641,90	17	760
VRPNC6	555,43	622,22	665,9	6	5
VRPNC7	909,68	955,76	1007,2	13*	10
VRPNC8	865,94	1013,71	1010,6	9	48
VRPNC9	1164,24	1322,19	1376,7	15*	162
VRPNC10	1403,21	-	1691,7	20*	559
VRPNC11	1042,11	1108,87	1190,1	7	562
VRPNC12	819,56	903,72	937,8	10	95
VRPNC13	1545,80	1686,65	1640,4	11	125
VRPNC14	866,37	920,29	951,7	11	74

* o sistema não resolveu o problema com o número de veículos sugerido. Os números de veículos sugeridos são 11,14 e 18, respectivamente (ver tabela 5).

O *software TransCAD* não resolveu os problemas VRPNC 7, 9 e 10 de forma adequada, pois a quantidade de veículos atribuída foi maior que a sugerida pelos autores, fato que pode ser observado pelas tabelas 5 e 8. Já o *software Delivery* resolveu todos os problemas testes deste conjunto com o número de veículos sugerido. Além disso, o *software Delivery* possibilita que o usuário simule o problema atribuindo a quantidade de veículos desejada. Para fins comparativos, o mesmo problema foi então executado neste sistema com as quantidades de veículos

¹ Valores encontrados em: PUREZA, V. M. (1990). *Problemas de roteamento de veículos via metaheurística tabu*. Tese de Mestrado. Faculdade de Engenharia Elétrica e de Computação - UNICAMP.

alvo (originalmente definida na criação dos problemas teste), com a quantidade de veículos sugerida pelo sistema como a ideal (que é a menor possível) e com a mesma quantidade de veículos que a utilizada pelo *software TransCAD*, visando diferentes níveis de análise.

Tabela 8: Soluções apresentadas pelo *software Delivery* para os problemas propostos por Christofides *et al.* (1979).

Problema	Melhor solução	Solomon (1987)	Delivery		
			Custo (km)	Número veículos	Tempo computacional (segundos)
VRPNC1	524,61	611,17	694,18	5*	11
VRPNC2	835,26	996,44	1231,83	10*	14
VRPNC3	826,14	1048,02	1039,60	8*	31
VRPNC4	1028,42	1297,72	1625,20	12*	65
VRPNC5	1300,89	-	2002,92	17*	98
VRPNC6	555,43	622,22	670,84	6*	9
VRPNC7	909,68	955,76	1144,03	11*	35
			1197,69	13	22
VRPNC8	865,94	1013,71	1020,36	8	53
			1001,14	9*	35
			1450,12	13	69
VRPNC9	1164,24	1322,19	1487,55	14*	69
			1457,49	15	75
			2002,92	17	340
VRPNC10	1403,21	-	1839,30	18*	157
			1751,88	20	113
			1649,45	7*	38
VRPNC11	1042,11	1108,87	1111,00	10*	48
VRPNC12	819,56	903,72	1852,00	11*	62
VRPNC13	1545,80	1686,65	1103,41	10	31
VRPNC14	866,37	920,29	1166,55	11*	34

* correspondem aos valores que devem ser comparados com a literatura, em função da quantidade de veículos utilizada.

No que diz respeito à qualidade da solução, a comparação consistiu em avaliar a performance dos algoritmos comparando as distâncias obtidas nas soluções apresentadas pelos *software Delivery* e *TransCAD*, com os resultados encontrados na literatura. O procedimento utilizado para comparar as heurísticas, foi o método estatístico de Wilcoxon. Os problemas que não foram resolvidos de forma adequada, em função da quantidade de veículos atribuída pelo *software* ser diferente da quantidade sugerida pelos autores na criação dos problemas teste (por uma das

ferramentas utilizadas), foram desconsiderados neste processo de análise. A tabela 9 traz uma primeira comparação entre os resultados apresentados pelas heurísticas do *TransCAD* e *Delivery*.

Tabela 9: Comparação das soluções apresentadas pelas heurísticas nos 14 problemas testes de Christofides et al.,1979. ($Razão(i)=Distância/Limite\ inferior(i)$).

Número do problema	Limite inferior 1 (melhor solução)	Limite inferior 2 (Solomon, 1987)	TransCAD			Delivery		
			Distância	Razão 1	Razão 2	Distância	Razão 1	Razão 2
VRPNC1	524,61	611,17	639,10	1,22	1,05	694,18	1,32	1,14
VRPNC2	835,26	996,44	985,70	1,18	0,99	1231,83	1,47	1,24
VRPNC3	826,14	1048,02	1017,70	1,23	0,97	1039,6	1,26	0,99
VRPNC4	1028,42	1297,72	1330,80	1,29	1,03	1625,2	1,58	1,25
VRPNC5	1300,89	-	1641,90	1,26	-	2002,92	1,54	-
VRPNC6	555,43	622,22	665,9	1,20	1,07	670,84	1,21	1,08
VRPNC7	909,68	955,76	-	-	-	1144,03	1,26	1,20
VRPNC8	865,94	1013,71	1010,6	1,17	0,99	1001,14	1,16	0,99
VRPNC9	1164,24	1322,19	-	-	-	1487,55	1,28	1,13
VRPNC10	1403,21	-	-	-	-	1839,30	1,31	-
VRPNC11	1042,11	1108,87	1190,1	1,14	1,07	1649,45	1,58	1,49
VRPNC12	819,56	903,72	937,8	1,14	1,04	1111,00	1,36	1,23
VRPNC13	1545,80	1686,65	1640,4	1,06	0,97	1852,00	1,20	1,10
VRPNC14	866,37	920,29	951,7	1,10	1,03	1166,55	1,35	1,27



Melhor desempenho

O desvio (em porcentagem) das soluções apresentadas pelos *software* aqui testados, em relação ao valores tomados como base de comparação, é calculado por $(razão_i - 1) * 100$, mostrados na tabela 10.

Nesta tabela, X_{1i} e X_{2i} correspondem as porcentagens desviadas das soluções apresentadas pelo *software Delivery* em relação às melhores soluções encontradas na literatura e em relação às soluções apresentadas pela heurística do Solomon (1987), respectivamente. Da mesma forma, Y_{1i} e Y_{2i} correspondem às porcentagens desviadas das soluções apresentadas pelo *software TransCAD*. O sinal negativo indica que o

resultado apresentado pelos *software* testados é melhor que aquele tomado como base de comparação.

Tabela 10: Porcentagens desviadas das soluções encontradas.

Problema	TransCAD (%)		Delivery (%)	
	Y_{1i}	Y_{2i}	X_{1i}	X_{2i}
VRPNC1	21,82	4,57	32,32	13,58
VRPNC2	18,01	-1,08	47,48	23,62
VRPNC3	23,19	-2,89	25,84	-0,80
VRPNC4	29,40	2,55	58,03	25,24
VRPNC5	26,21	-	53,97	-
VRPNC6	19,89	7,02	20,78	7,81
VRPNC7	-	-	25,76	19,70
VRPNC8	16,71	-0,31	15,61	-1,24
VRPNC9	-	-	27,77	12,51
VRPNC10	-	-	31,08	-
VRPNC11	14,20	7,33	58,28	48,75
VRPNC12	14,43	3,77	35,56	22,94
VRPNC13	6,12	-2,74	19,81	9,80
VRPNC14	9,85	3,41	34,65	26,76
Média	18,16	1,97	34,78	17,39

A tabela 11 mostra que a heurística do *TransCAD* apresenta, na média, desempenho melhor que a heurística do *Delivery*, quando aplicados sobre problemas com as características aqui levadas em consideração. No entanto, foi feita uma análise estatística para aferir o nível de significância do resultado desta comparação. A tabela 12 dá início a aplicação da técnica estatística utilizada, através do cálculo do fator de Wilcoxon (W).

Tabela 11: Heurística do *Delivery* x heurística do *TransCAD*.

	Delivery	TransCAD
Número de vezes em que uma heurística apresenta resultado melhor sobre a outra	4	10
Porcentagem média acima do limite inferior 1*	34,78%	18,16%
Porcentagem média acima do limite inferior 2*	17,39%	1,97%
Pior desempenho em porcentagem acima do limite inferior 1	58,28%	29,40%
Pior desempenho em porcentagem acima do limite inferior 2	48,75%	7,33%

* ver média na tabela 10.

Tabela 12: Comparação da performance das heurísticas.

X_{1i}	Y_{1i}	$X_{1i}-Y_{1i}$	Rank de $ X_{1i}-Y_{1i} $	R_{1i}	X_{2i}	Y_{2i}	$ X_{2i}-Y_{2i} $	Rank de $ X_{2i}-Y_{2i} $	R_{2i}
32,32	21,82	10,50	4	4	13,58	4,57	9,01	4	4
47,48	18,01	29,47	10	10	23,62	-1,08	24,70	10	10
25,84	23,19	2,65	3	3	-0,80	-2,89	2,09	3	3
58,03	29,40	28,63	9	9	25,24	2,55	22,69	8	8
53,97	26,21	27,75	8	8	-	-	-	-	-
20,78	19,89	0,89	1	1	7,81	7,02	0,79	1	1
25,76	-	-	-	-	19,70	-	19,70	7	7
15,61	16,71	-1,09	2	-2	-1,24	-0,31	-0,93	2	-2
27,77	-	-	-	-	12,51	-	-	-	-
31,08	-	--	-	-	-	-	-	-	-
58,28	14,20	44,08	11	11	48,75	7,33	41,43	11	11
35,56	14,43	21,13	6	6	22,94	3,77	19,17	6	6
19,81	6,12	13,69	5	5	9,80	-2,74	12,54	5	5
34,65	9,85	24,80	7	7	26,76	3,41	23,35	9	9
W = 62					W = 62				

Os resultados apresentados na tabela 11 indicam que é mais apropriado aplicar o teste de Wilcoxon unicaldal através das seguintes hipótese:

H_0 : a heurística do *Delivery* apresenta desvio igual ao da heurística do *TransCAD*
($E[X] = E[Y]$);

H_1 : a heurística do *Delivery* apresenta desvio maior que a heurística do *TransCAD*
($E[X] > E[Y]$).

A hipótese nula ($E[X] = E[Y]$) deverá ser rejeitada com um nível de significância de 5%, se $W > W_{0,95}$. Utilizando a formula (seção 5.5) proposta por Golden e Stewart (1985) para a estimativa de $W_{1-\alpha}$, tem-se que:

$$W_{0,95} (n=10) = 32,27$$

$$W_{0,95} (n=11) = 37$$

Assim, a hipótese nula é rejeitada com nível de significância de 0,05, de onde se pode inferir que os desvios das soluções apresentadas pelo *software Delivery* em relação a melhor solução, são maiores que os desvios das soluções apresentadas pelo *software TransCAD*. Na realidade, a hipótese nula é rejeitada com um nível de significância menor que 0,02, onde $W_{0,98} (n=10) = 46,11$ e $W_{0,98} (n=11) = 40,22$. Assim, conclui-se que o *software TransCAD* apresenta soluções de melhor qualidade que o *software Delivery*, com um nível de significância de 0,02.

Uma análise mais geral é conduzida na tabela 13, onde valendo-se de estatística descritiva, comparou-se as soluções apresentadas pelos *software Delivery* e *TransCAD*. Os testes foram executados várias vezes no *software Delivery* priorizando: 1. A minimização da distância total (independente da frota utilizada); 2. A minimização da frota; 3. A utilização da mesma quantidade de veículos utilizada pelo *software TransCAD*; e 4. Utilização da frota de veículos sugerida pelos autores na criação dos problemas testes. Os resultados de “*TransCAD*” dizem respeito à média dos valores apresentados por este sistema para todos os problemas, independente da frota proposta.

Tabela 13: Comparação geral do desempenho dos *software Delivery* e *TransCAD*.

	Distância		Número médio de veículos		Tempo médio de processamento (segundos)	
	Média	Desvio padrão	Média	Desvio padrão	Média	Desvio padrão
Melhor solução	978,00	295,62	10,64	3,77	-	-
Solomon (1987)	1040,56	296,96	9,50	2,61	-	-
<i>TransCAD</i>	1149,09	342,57	11,00	4,21	213,79	255,27
<i>Delivery</i> 1	1309,11	403,16	10,64	3,90	47,07	30,12
<i>Delivery</i> 2	1328,42	426,29	10,36	3,47	64,57	80,07
<i>Delivery</i> 3	1317,98	399,98	11,00	4,21	46,78	30,93
<i>Delivery</i> 4	1322,54	409,53	10,64	3,64	50,43	37,98

O interessante desta análise é notar que o procedimento *default* para roteirização de veículos no *software Delivery* - "*Delivery* 2" na tabela 13 - representa o pior desempenho médio da heurística, no que diz respeito à qualidade de solução, robustez e tempo de processamento. O procedimento *default* desta heurística prioriza a diminuição da frota de veículos. Nas quatro diferentes formas de manipular o problema, todas elas apresentaram resultados que na média são piores que as soluções apresentadas pelo *software TransCAD*, salvo o tempo de processamento, onde o *software Delivery* apresenta menor tempo computacional.

As figuras 11 e 12 mostram a robustez dos *software Delivery* e *TransCAD*, aferida em função do tamanho e características do problema (a relação entre o número do problema e suas características é feita na tabela 5) e das porcentagens desviadas em relação as bases de comparação. A interpretação se dá da seguinte

forma: quanto menor a oscilação em torno do eixo vertical, maior a robustez da heurística do sistema.

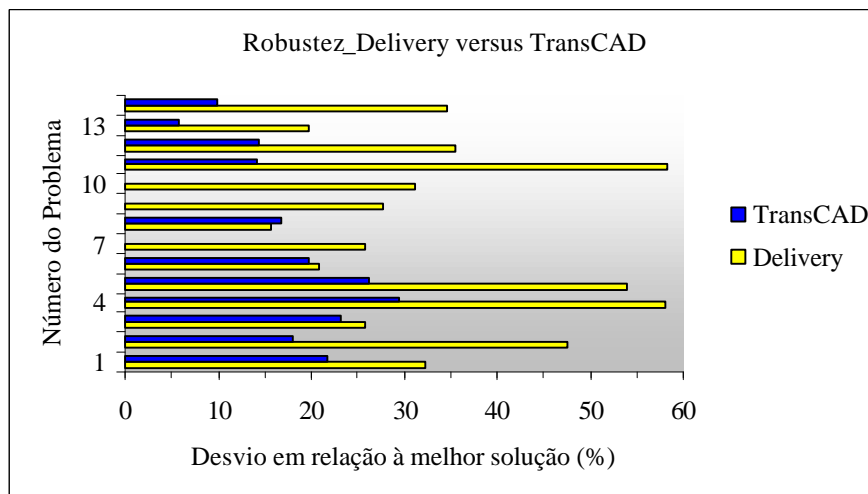


FIGURA 11– ROBUSTEZ DOS SOFTWARE *DELIVERY* E *TRANS*CAD EM RELAÇÃO À MELHOR SOLUÇÃO.

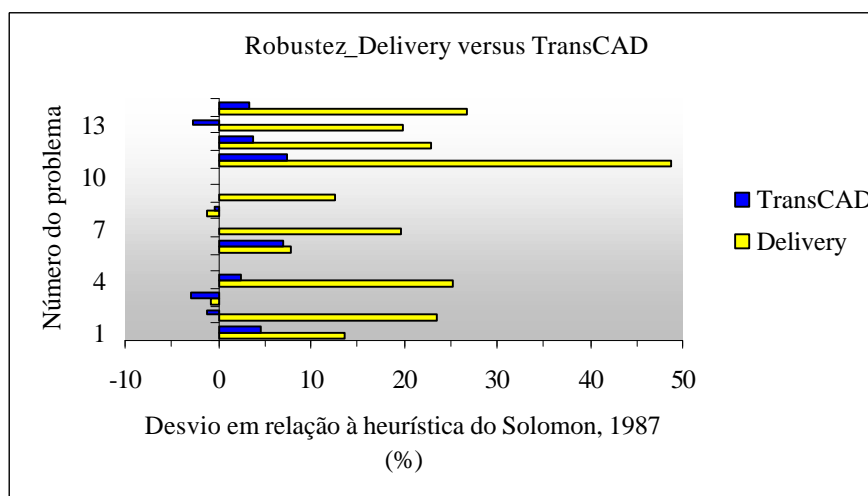


FIGURA 12– ROBUSTEZ DOS SOFTWARE *DELIVERY* E *TRANS*CAD EM RELAÇÃO À HEURÍSTICA DO SOLOMON.

O tempo computacional foi analisado relacionando os tempos de processamento apresentados pelos *software Delivery* e *TransCAD* em função do tamanho do problema (número de clientes) e também em função da razão *número de clientes/quantidade de veículos*. Encontra-se o resumo desta análise na tabela 14.

Tabela 14: Análise do Tempo Computacional dos *software Delivery* e *TransCAD*.

Variáveis	Função	Delivery	TransCAD
Tempo de processamento - em segundos (t) versus número de clientes (n)	Linear	$t = 0,6589n - 27,957$ $R^2 = 0,9442$	$t = 4,5895n - 306,79$ $R^2 = 0,7242$
	Exponencial	$t = 6,2033e^{0,0156n}$ $R^2 = 0,8755$	$t = 2,4369e^{0,0012n}$ $R^2 = 0,8092$
	Polinomial (grau 3)	$t = -10^{-5}n^3 + 0,0053n^2 - 0,0517n - 0,1283$ $R^2 = 0,9492$	$t = 0,0002n^3 + 0,0727n^2 - 5,7817n + 120,46$ $R^2 = 0,7495$
	Polinomial (grau 2)	$t = 0,0009n^2 + 0,4377n - 16,009$ $R^2 = 0,9482$	$t = 0,0167n^2 + 0,38n - 79,478$ $R^2 = 0,7472$
	Logarítmica	$t = 68,891\ln(n) - 273,37$ $R^2 = 0,8659$	$t = 466,24\ln(n) - 1953$ $R^2 = 0,6271$
	Potência	$t = 0,0099n^{1,7651}$ $R^2 = 0,9438$	$t = 6 * 10^{-6} * n^{3,5493}$ $R^2 = 0,8807$
Tempo de processamento - em segundos (t) versus número de clientes/número de veículos (n)	Linear	$t = 3,106n + 14,281$ $R^2 = 0,0671$	$t = 56,637n - 378,93$ $R^2 = 0,353$
	Exponencial	$t = 11,343e^{0,1111n}$ $R^2 = 0,1427$	$t = 1,0902e^{0,4146n}$ $R^2 = 0,4584$
	Polinomial (grau 3)	$t = -0,1326n^3 + 3,421n^2 - 19,841n + 38,269$ $R^2 = 0,2374$	$t = -1,1022n^3 + 37,485n^2 - 340,28n + 927,41$ $R^2 = 0,3695$
	Polinomial (grau 2)	$t = -1,1174n^2 + 28,586n - 122,55$ $R^2 = 0,2224$	$t = -0,2524n^2 + 62,392n - 409,83$ $R^2 = 0,3531$
	Logarítmica	$t = 40,557\ln(n) - 47,223$ $R^2 = 0,1054$	$t = 576,29\ln(n) - 1122$ $R^2 = 0,3364$
	Potência	$t = 1,7461n^{1,009}$ $R^2 = 0,1822$	$t = 0,0028n^{4,446}$ $R^2 = 0,4851$

Quanto maior o valor de R-quadrado (tabela 14) melhor a função representa a complexidade computacional dos *software*. Assim, é claramente perceptível que o tempo computacional nos *software Delivery* e *TransCAD* cresce em função do número de clientes. A função de complexidade computacional que melhor representa o crescimento do tempo de processamento no *software TransCAD*, consiste em uma função de potência. Já no *software Delivery* a função complexidade computacional mais adequada para representar o comportamento do seu tempo de processamento consiste em uma função polinomial de grau 3, embora as funções linear e polinomial de grau 2 também apresentem um alto coeficiente de determinação. Neste caso, a diferença entre os coeficientes de R^2 é tão pequena que fica difícil afirmar que a função de complexidade computacional que melhor representa o tempo de processamento do *software* é de fato a polinomial de grau 3, principalmente dado o pequeno número de casos estudados. As figuras 13 e 14 demonstram os gráficos com estes resultados. É importante dizer que estas funções são válidas para representar o tempo computacional em casos semelhantes a estes estudados, de forma que as

características do problema e o número de clientes considerados pertençam ao intervalo aqui especificado.

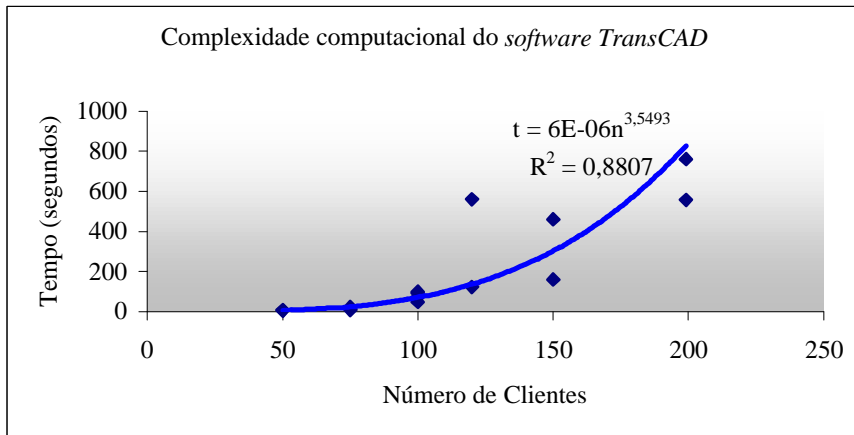


FIGURA 13- GRÁFICO REPRESENTANDO O CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DO NÚMERO DE CLIENTES NO *SOFTWARE TRANSCAD*, BEM COMO A FUNÇÃO QUE MELHOR REPRESENTA A SUA COMPLEXIDADE COMPUTACIONAL.

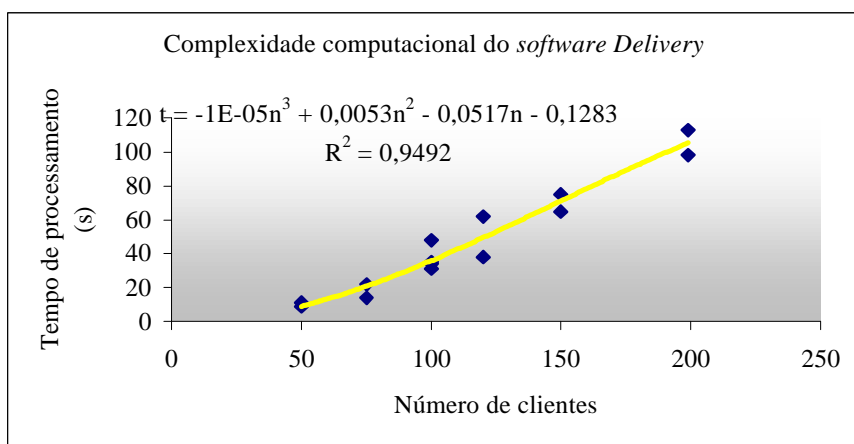


FIGURA 14 - GRÁFICO REPRESENTANDO O CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DO NÚMERO DE CLIENTES NO *SOFTWARE DELIVERY*, BEM COMO A FUNÇÃO QUE MELHOR REPRESENTA A SUA COMPLEXIDADE COMPUTACIONAL.

7.4.2 Problemas testes de Solomon (1987)

Os *software* testados não resolvem problemas com janela de tempo que ultrapassem um período de 24 horas. Em função disso, não foi possível executar os conjuntos de problemas teste C2, RC2 e R2.

O *software Delivery* possui uma grande limitação com relação a estrutura do banco de dados para cadastramento das janelas de tempo nos clientes. As janelas de tempo dos clientes são definidas em função de horas "inteiras" do dia, como por exemplo, entre 7 e 8 horas. Ou seja, não é possível definir uma janela de tempo de um cliente menor que uma hora. Assim, durante a formulação dos problemas testes neste *software*, quando há um cliente com uma janela de tempo definida entre 9h23min e 10h05min, por exemplo, o horário desta janela de tempo deverá ser cadastrada como o horário compreendido entre 9 e 11 horas.

Em virtude deste fato, ao formular estes problemas no *software Delivery*, as janelas de tempo foram definidas de forma a limitar o máximo possível o horário de atendimento em torno da janela de tempo dos clientes especificada na criação dos problemas testes. Este procedimento compromete a característica inicial dos problemas, podendo favorecer ou prejudicar o *software Delivery* ao comparar suas soluções neste processo de análise empírica comparativa. Como uma forma de manter a imparcialidade na comparação entre os dois *software*, estas mesmas janelas de tempo definidas no *software Delivery* foram reproduzidas nos testes com o *software TransCAD*. Além disso, também foram executados no *software TransCAD* os problemas testes com as janelas de tempo dos problemas originais, definidas durante a sua criação. Tais resultados estão mostrados nas tabelas 15 e 16.

Os resultados do *software TransCAD versus* a melhor solução encontrada na literatura foram comparados inicialmente. Depois de aferido o desempenho do *software TransCAD*, foi aferido o desempenho do *software Delivery* em relação ao *software TransCAD*.

Tabela 15: Soluções apresentadas pelo *software TransCAD* para os problemas propostos por Solomon (1987).

Conjunto	Problema	TransCAD (JT original)			TrasCAD (JT = Delivery)		
		Custo	Número veículos	Tempo computacional (s)	Custo	Número veículos	Tempo computacional (s)
C1	C101	922,1	10	59	864,1	10	10
	C102	1027,0	10	22	1965,5	11	17
	C103	1224,6	11	31	988,4	11	16
	C104	1148,3	10	47	1148,3	10	42
	C105	864,1	10	15	959,50	11	15
	C106	996,1	11	17	998,0	10	17
	C107	901,8	10	22	958,0	11	20
	C108	974,4	10	22	960,4	11	27
	C109	933,0	11	42	935,0	11	38
Média		999,04	10,33	30,77	1086,35	10,66	22,44
RC1	RC101	1921,70	17	5	1590,40	13	12
	RC102	1818,50	15	10	1486,90	12	14
	RC103	1540,40	12	17	1302,30	11	20
	RC104	1365,00	12	28	1301,3	11	24
	RC105	1853,60	17	10	1443,3	12	14
	RC106	1726,70	14	11	1502,0	12	16
	RC107	1577,60	13	15	1374,9	11	17
	RC108	1374,50	11	22	1175,10	10	31
Média		1647,25	13,88	14,75	1397,03	11,50	18,50
R1	R101	1863,4	21	8	1300,50	13	11
	R102	1717,0	19	10	1316,20	12	15
	R103	1530,1	16	15	1146,10	11	21
	R104	1277,2	12	23	1357,90	11	28
	R105	1619,3	15	10	1346,90	12	11
	R106	1471,4	15	12	1269,80	11	15
	R107	1344,0	12	19	1203,30	11	22
	R108	1210,1	10	30	1130,00	10	23
	R109	1431,9	14	11	1291,00	11	17
	R110	1340,2	12	17	1186,80	11	22
	R111	1294,5	12	18	1185,20	11	23
	R112	1270,4	11	27	1069,40	10	35
Média		1447,46	14,08	16,67	1233,59	11,17	20,25

Tabela 16: Soluções apresentadas pelo *software Delivery* para os problemas propostos por Solomon (1987).

Conjunto	Problema	Delivery		
		Custo	Número veículos	Tempo computacional (s)
C1	C101	2907,26	16	45
	C102	2719,26	20	43
	C103	1893,22	11	43
	C104	1629,33	10	37
	C105	2984,48	16	45
	C106	2318,75	14	44
	C107	2412,71	13	43
	C108	2449,98	13	42
	C109	1979,29	12	39
Média		2366,03	13,89	42,33
RC1	RC101	1871,60	15	56
	RC102	1891,96	15	54
	RC103	1815,72	14	51
	RC104	1570,90	11	43
	RC105	1923,5	15	55
	RC106	1739,5	13	48
	RC107	1515,48	12	44
	RC108	1462,94	12	44
Média		1723,95	13,37	49,37
R1	R101	2098,72	8	54
	R102	2007,00	8	61
	R103	1812,27	8	66
	R104	1699,39	8	63
	R105	1992,00	8	57
	R106	1895,88	8	59
	R107	1683,33	8	55
	R108	1520,30	8	67
	R109	1936,33	8	57
	R110	1990,54	8	63
	R111	1696,86	8	58
	R112	1462,42	8	65
Média		1816,25	8,00	60,42

A tabela 17 traz a comparação entre as médias dos resultados destes testes aplicados nos *software Delivery* e *TransCAD*. Nesta tabela, "desvio_1" diz respeito a porcentagem desviada em relação à melhor solução encontrada na literatura revisada, quando comparada às soluções apresentadas pelo *software TransCAD* utilizando a janela de tempo proposta na criação dos testes. Já "desvio_2" corresponde à porcentagem desviada da solução encontrada pelo *software Delivery* em relação ao *software TransCAD*, utilizando a mesma janela de tempo. Assim, a qualidade da solução no *software TransCAD* será aferida com relação a solução ótima, enquanto o *software Delivery* terá sua qualidade avaliada em relação ao *software TransCAD*.

Tabela 17: Comparação entre os software *Delivery* e *TransCAD*

Conjunto		Melhor Solução *	TransCAD	Desvio_1 (%)	TransCAD (JT=Delivery)	Delivery	Desvio_2 (%)
C1	C101	828,38	922,1	11,31	864,10	2907,26	236,45
	C102	828,38	1027	23,98	1965,50	2719,26	38,35
	C103	828,38	1224,6	47,83	988,40	1893,22	91,54
	C104	828,38	1148,3	38,62	1148,30	1629,33	41,89
	C105	828,38	864,1	4,31	959,50	2984,48	211,05
	C106	828,38	996,1	20,25	998,00	2318,75	132,34
	C107	828,38	901,8	8,86	958,00	2412,71	151,85
	C108	828,38	974,4	17,63	960,40	2449,98	155,10
	C109	828,38	933	12,63	935,00	1979,29	111,69
Média	Distância	828,38	999,04	20,60%	1086,36	2366,03	130,03%
	No. de rotas	10	10,33	3,30%	10,67	13,89	30,30%
	Tempo computacional (s)	-	30,77		22,44	42,33	132,42%
RC1	RC101	1367,51	1921,70	40,53	1590,40	1871,60	17,68
	RC102	1367,51	1818,50	32,98	1486,90	1891,96	27,24
	RC103	1367,51	1540,40	12,64	1302,30	1815,72	39,42
	RC104	1367,51	1365,00	-0,18	1301,30	1570,90	20,72
	RC105	1367,51	1853,60	35,55	1443,30	1923,50	33,27
	RC106	1367,51	1726,70	26,27	1502,00	1739,50	15,81
	RC107	1367,51	1577,60	15,36	1374,90	1515,48	10,22
	RC108	1367,51	1374,50	0,51	1175,10	1462,94	24,49
Média	Distância	1367,51	1647,25	20,46%	1397,03	1723,95	23,61%
	No. de rotas	11,88	13,88	16,79%	11,50	13,37	16,26%
	Tempo computacional (s)	-	14,75		18,50	49,38	197,52%
R1	R101	1197,42	1863,40	55,62	1300,50	2098,72	61,38
	R102	1197,42	1717,00	43,39	1316,20	2007,00	52,48
	R103	1197,42	1530,10	27,78	1146,10	1812,27	58,12
	R104	1197,42	1277,20	6,66	1357,90	1699,39	25,15
	R105	1197,42	1619,30	35,23	1346,90	1992,00	47,90
	R106	1197,42	1471,40	22,88	1269,80	1895,88	49,31
	R107	1197,42	1344,00	12,24	1203,30	1683,33	39,89
	R108	1197,42	1210,10	1,06	1130,00	1520,30	34,54
	R109	1197,42	1431,90	19,58	1291,00	1936,33	49,99
	R110	1197,42	1340,20	11,92	1186,80	1990,54	67,72
	R111	1197,42	1294,50	8,11	1185,20	1696,86	43,17
	R112	1197,42	1270,40	6,09	1069,40	1462,42	36,75
Média	Distância	1197,42	1147,46	20,88%	1233,59	1816,25	47,20%
	No. de rotas	11,88	14,08	11,95%	11,17	8	-28,00%
	Tempo computacional (s)	-	16,67		20,25	60,42	229,10%

* melhor solução em termos de distância média percorrida (retirada da tabela 6).

Assim, os resultados deste conjunto de testes permitem extrair as seguintes conclusões:

- 1) Para o caso de problemas onde os clientes se encontram geograficamente agrupados e possuem restrições de janela de tempo (C1), o *software TransCAD* apresenta desempenho médio de 20,60% pior que a "melhor solução" (apresentando desvio máximo e mínimo de 47,83% e 4,31%, respectivamente) no que diz respeito à qualidade da solução (distância total percorrida). O número médio de rotas é de 3,3% maior que o número médio de rotas da melhor solução (desvio máximo e mínimo da solução ótima de 10% e 0%, respectivamente). Já o *software Delivery* demonstra um desempenho médio na qualidade da solução de 130,03% pior que o *software TransCAD* (variando entre 38,35% e 236,45%), com um número médio de rotas 30,3% maior que este último (variando entre 9,09% e 81,82%). O tempo de processamento médio no *software Delivery* é 132,42% maior que no *software TransCAD* (variando entre 11,90% menor que o tempo no *software TransCAD* e 350,00% maior que no *TransCAD*). Cabe observar, no entanto, que em nenhum caso estudado este tempo ultrapassou os 45 segundos.
- 2) No caso de problemas onde os clientes encontram-se dispersos uniformemente em um plano geográfico e possuem restrições de janela de tempo (R1), o *software TransCAD*, na média, apresenta soluções com uma distância total percorrida 20,88% maior que a ótima (onde o pior resultado foi um desvio de 55,62% maior que a melhor solução e o melhor resultado foi um desvio de 1,06% também maior), com um número médio de rotas 11,95% maior que a solução ótima (sendo que no melhor caso ele diminuiu o número de rotas em 22,51% da melhor solução, e no pior representou um aumento de 66,93% do número de rotas). Já o *software Delivery* demonstra desempenho médio na qualidade da solução de 47,20% pior que o *software TransCAD* (desviando entre 25,15% e 67,72% para pior nas soluções apresentadas pelo *TransCAD*), no entanto, consegue uma diminuição de 28% no número médio de rotas apresentado pelo *software TransCAD* (diminuindo no máximo 38,46% e no mínimo 20%). O tempo de processamento médio no *software Delivery* é 229,10% maior que no *software TransCAD* (no mínimo 85,71% maior e

no máximo 418,18% maior). Tal como observado no item anterior, é importante destacar que o tempo de processamento em nenhum destes casos ultrapassou 67 segundos.

- 3) Problemas, nos quais existe uma mistura de clientes geograficamente agrupados e uniformemente dispersos, com restrição de janela de tempo (RC1) o *software TransCAD* apresenta, na média, soluções com distância total 20,46% maior que a ótima (variando entre 0,18% melhor e 40,53% pior que a "solução ótima") e um número médio de rotas 16,79% maior que a ótima. O *software Delivery* apresenta soluções com qualidade que na média são: 23,61% pior que o *software TransCAD* (variando entre 10,22% e 33,27% para pior), com número médio de rotas 16,26% maior que no *software TransCAD* (no seu melhor desempenho iguala – RC104, e no pior atribui um número de rotas 27,27% maior). O tempo de processamento médio no *software Delivery* é 197,52% maior que o *software TransCAD* (variando entre 41,94% e 366,67%).

7.4.3 Conjunto de problemas com dados reais

Os resultados encontram-se mostrados na tabela 18.

Para efeito de comparação dos resultados, os problemas 1 e 2 (ver item 7.3) são exatamente os mesmos. A diferença nas soluções apresentadas consiste na maneira como as informações foram manipuladas pelo *software* em questão. No primeiro problema, foram pré definidos os clientes que cada veículo deveria atender e pediu-se para os sistemas roteirizá-los. No segundo problema, coube aos sistemas escolher os pontos a serem atendidos por cada veículo, de forma que o número de clientes atendidos fosse igual ao do problema 1.

Tabela 18: Soluções apresentadas pelos *software Delivery* e *TransCAD* para o conjunto de problemas com dados reais.

Problema	Rota	Empresa		TransCAD			Delivery		
		Tempo ciclo	Distância (Km)	Tempo processamento	Tempo ciclo	Distância (Km)	Tempo processamento	Tempo ciclo	Distância (Km)
1	Perua	3h06m	22,96		1h15m	18,20		3h38m	21,28
	Caminhão	6h12m	66,78	2'07"	5h43m	47,30	4'23"	5h48m	75,04
2	Perua	3h06m	22,96		2h01m	13,40		2h57m	41,02
	Caminhão	6h12m	66,78	6'32"	6h01m	50,90	5'25"	6h02m	77,52
3	Perua	-	-		1h09m	14,40		3h02m	20,13
	Caminhão	-	-	2'09"	5h45m	38,70	26"	4h35m	62,53
4	Perua	-	-		1h59m	10,90		2h52m	19,70
	Caminhão	-	-	6'43"	3h47m	39,20	23"	5h03m	68,81
5	Perua	-	-		1h15m	18,20		1h22m	22,17
	Caminhão	-	-	4'51"	3h42m	46,40	4'14"	3h51m	52,94
6	Perua	-	-		44m	10,40		1h15m	18,34
	Caminhão	-	-	7'10"	4h07m	50,80	4'18"	4h02m	58,48
7	Perua	-	-		1h09m	14,40		55m	14,96
	Caminhão	-	-	4'22"	3h26m	37,30	20"	2h57m	47,12
8	Perua	-	-		40m	8,40		44m	15,12
	Caminhão	-	-	8'45"	3h47m	39,20	19"	2h22m	46,31

A tabela 19 traz uma comparação dos resultados apresentados na tabela 18. A comparação foi feita através da porcentagem desviada da solução apresentada pela empresa (tempo de ciclo por rota e total; distâncias percorridas por rota e total) em relação às soluções apresentadas pelos *software* - $[(\text{solução apresentada pelo software}/\text{solução apresentada pela empresa}) - 1] * 100$. Assim, o sinal negativo indica que o *software* apresentou uma solução melhor que a sugerida pela empresa. O tempo de ciclo da perua fornecido pela empresa não será utilizado como um referencial de comparação. Tal veículo é operado pelo dono da empresa que, visando conquistar a fidelidade de seus clientes e assegurar a fatia de mercado, eleva o tempo de serviço nos pontos de atendimento de forma aleatória, tornando difícil a sua modelagem em nível de programação da rota. Este fato prejudica a comparação *empresa x sistemas* (vale ressaltar que é apenas para o tempo de ciclo e não distância percorrida), mas não interfere na comparação entre os resultados fornecidos pelas duas ferramentas já que foi assumido um tempo de serviço igual a 2 minutos para todos os estabelecimentos.

Nos outros problemas (3 a 8), criados a partir dos dados reais mas que não consistem em problemas reais, a performance dos algoritmos dos *software* testados

foi comparada através do desvio (%) relativo entre o *software Delivery* e *TransCAD*, dado pela relação $[(\text{solução apresentada pelo software Delivery}/\text{solução apresentada pelo software TransCAD}) - 1] * 100$. Desta forma, quando há o sinal negativo significa que o *software Delivery* conseguiu um resultado melhor que o *software TransCAD*.

Os desvios relativos entre os *software Delivery* e *TransCAD* representaram objeto de análise de desempenho do *software Delivery*. Partiu-se do princípio de que o *software TransCAD* representa uma ferramenta robusta e confiável, como vem demonstrando ser ao longo deste processo de análise, e principalmente, em função dos resultados bastante equilibrados apresentados nas soluções deste último conjunto de testes (observar tabela 18).

Tabela 19: Comparação do desempenho dos *software Delivery* e *TransCAD* em uma situação real.

Problema	Rota	Empresa x TransCAD		Empresa x Delivery		Delivery x TransCAD		
		Tempo ciclo (%)	Distância (%)	Tempo ciclo (%)	Distância (%)	Tempo ciclo (%)	Distância (%)	Tempo processamento (%)
1	perua	-59,68	-20,73	17,20	-7,32	190,67	16,92	
	caminhão	-7,80	-29,17	-6,45	12,37	1,46	58,65	
	GLOBAL	-25,09	-27,01	1,43	7,33	35,41	47,05	107,09
2	perua	-34,95	-41,64	-4,84	78,66	46,28	206,12	
	caminhão	-2,96	-23,78	-2,69	16,08	0,28	52,30	
	GLOBAL	-13,62	-28,35	-3,41	32,09	11,83	84,35	-17,09
3	perua					163,77	39,79	
	caminhão					-20,29	61,58	
	GLOBAL					10,39	55,67	-79,36
4	perua					44,54	80,73	
	caminhão					33,48	75,54	
	GLOBAL					37,28	76,67	-94,29
5	perua					9,33	21,81	
	caminhão					4,05	14,09	
	GLOBAL					5,39	16,27	-12,71
6	perua					70,45	76,35	
	caminhão					-2,02	15,12	
	GLOBAL					8,93	25,52	-40,00
7	perua					-20,29	3,89	
	caminhão					-14,08	26,33	
	GLOBAL					-15,64	20,08	-92,37
8	Perua					10,00	80,00	
	caminhão					-37,44	18,14	
	GLOBAL					-30,34	29,05	-96,38

De um modo geral todos estes problemas são os mesmos: existem 92 clientes que devem ser atendidos por dois veículos de diferentes capacidades, partindo de um único depósito. Além disso, três fatores são explorados: a presença de uma rede viária; o pré agrupamento dos clientes em função do veículo que deverá atendê-lo (ou definição de clientes por região), e restrições de janela de tempo em alguns clientes. Todas estas características estão presentes, simultaneamente, no problema de número 1. Tendo este como o "problema básico", os outros foram criados através de pequenas variações, onde os parâmetros (ou fatores) foram desconsiderados um a um, com a intenção de reconhecer qual deles traz um maior impacto no desempenho do algoritmo. Colocando de uma outra maneira, pretende-se com este conjunto de testes avaliar qual a sensibilidade do algoritmo em relação a presença ou não dos fatores acima relacionados, além de avaliar o desempenho destes *software* quando aplicados em situações reais. Os fatores presentes em cada um destes problemas estão relacionados no quadro 9.

Quadro 9: Fatores explorados em cada problema

Número do problema	Janela de tempo	Rede viária	Pré-agrupamento dos clientes
1	Sim	Sim	Sim
2	Sim	Sim	Não
3	Sim	Não	Sim
4	Sim	Não	Não
5	Não	Sim	Sim
6	Não	Sim	Não
7	Não	Não	Sim
8	Não	Não	Não

Os principais pontos aqui analisados possibilitaram as seguintes conclusões:

- 1) Quando foi considerado o pré agrupamento dos clientes por veículo na formulação do problema, utilizando o procedimento da empresa para agrupamento de clientes por região, o *software TransCAD* apresentou uma redução de 7,80% no tempo de ciclo do caminhão e de 27,01% na distância total percorrida, na atividade relacionada à distribuição física da empresa em questão. Já o *software Delivery* não apresentou um bom desempenho, pois obteve um resultado pior do que o apresentado por método intuitivo pelos funcionários encarregados pela atividade de distribuição da empresa em questão. Apesar deste sistema conseguir uma

redução de 6,45% no tempo de ciclo do caminhão, apresentou um aumento de 7,33% na distância total percorrida. Assim, o *software Delivery* apresentou um desempenho pior que o *software TransCAD*, de forma que o tempo de ciclo apresentado pelo primeiro foi 35,41% maior que o apresentado pelo segundo e a distância total percorrida foi de 47,05% maior.

- 2) Quando não foi considerado o pré agrupamento dos clientes por veículo, o *software TransCAD* diminuiu os custos de transporte da empresa, através da redução de 2,96% no tempo de ciclo do caminhão e, de 28,35% na distância total percorrida (contra uma redução de 6,45% e 27,01%, respectivamente, quando os clientes foram pré agrupados). Já o *software Delivery* apresentou uma redução de 2,69% no tempo gasto pela empresa para realizar a rota do caminhão. No entanto, a distância total apresentada pelo sistema foi de 32,09% maior que a percorrida pelos veículos da empresa (contra uma diminuição de 6,45% no tempo de ciclo do caminhão e um aumento de 7,33% na distância total, quando os clientes foram agrupados). Assim, o *software Delivery* apresentou um desempenho pior que o *software TransCAD*, de forma que o tempo de ciclo apresentado pelo primeiro foi 11,83% maior que o apresentado pelo segundo e a distância total percorrida foi de 84,35% maior.

A comparação entre as soluções apresentadas pelos sistemas quando se considera ou não o pré agrupamento dos clientes, pode ser um indicativo da robustez das suas heurísticas. Ou seja, a heurística do *software TransCAD* é menos sensível (e portanto mais robusta) à restrição de pré agrupamento, quando comparada com a heurística do *software Delivery*. Os resultados indicaram claramente o quanto a heurística do *software Delivery* se mostra perturbada na presença ou ausência dos fatores aqui considerados. Esta análise será mais pormenorizada nos itens que seguem.

- 1) A heurística do *software Delivery* é muito sensível, no sentido de piorar as soluções, quando são consideradas as janelas de tempo dos clientes. Este *software* não funciona adequadamente nesta situação, fato que pode ser

notado durante a transição do problema 1 para o problema 5. Enquanto o problema 1 (com janela de tempo) apresentou um desvio de 47,05% (distância) e 35,41% (tempo de ciclo) para pior em relação à solução apresentada pelo *software TransCAD*, no problema 5 (sem janela de tempo) estes desvios diminuem para 16,27% (distância) e 5,39% (tempo), demonstrando uma maior aproximação na qualidade das soluções apresentadas entre as heurísticas comparadas. Quando os clientes não são pré agrupados por veículo, a heurística do *software Delivery* apresenta resultados com qualidade questionável. Comparando os desvios nos problemas 1 e 2 nota-se que, quando não houve o pré agrupamento dos clientes, o algoritmo prioriza muito a diminuição do tempo de ciclo penalizando a distância percorrida. Provavelmente, o que acontece é que quando o algoritmo analisa a viabilidade de inserção de um cliente estrategicamente posicionado durante a construção da rota, e este encontra-se em horário fora da sua janela de tempo (janela de tempo "fechada"), o algoritmo prioriza outro cliente com janela de tempo "aberta", sem ponderar o custo/benefício de permanecer com o veículo esperando por alguns minutos para "abertura" da sua janela de tempo. Comparando ainda os problemas 5 e 6 pode-se confirmar que o algoritmo do *Delivery* funciona melhor pré agrupando os clientes por região.

- 2) Aparentemente, o algoritmo do *software Delivery* está melhor preparado para trabalhar com redes viárias do que sem, fato que pode ser observado comparando-se os desvios nos problemas 6 (com rede) e 8 (sem rede), apesar da presença da rede aumentar significativamente o tempo computacional. No problema 6, onde considera-se a rede viária, os desvios relativos entre os *software Delivery e TransCAD* se mostram mais equilibrados do que os desvios relativos no problema 8.
- 3) Tempo de Processamento: o *software Delivery* apresenta, no geral, um tempo de processamento menor que o do *software TransCAD*, como pode ser notado nos resultados apresentados na tabela 18. O quadro 10 traz a relação crescente dos tempos de processamento em função dos fatores

presentes ou ausentes nos problemas teste, para o *software Delivery*. O mesmo foi feito para o *software TransCAD* no quadro 11.

Quadro 10: Análise do tempo de processamento no *software Delivery*.

Ordem de crescimento	Número do Problema	Tempo de processamento (s)	Características (fatores presentes e ausentes)
1	8	19	s/ JT; s/ Agrupamento; s/ Rede
2	7	20	s/ JT; c/ Agrupamento; s/ Rede
3	4	23	c/ JT; s/ Agrupamento; s/ Rede
4	3	26	c/ JT; c/ Agrupamento; s/ Rede
5	5	254	s/ JT; c/ Agrupamento; c/ Rede
6	6	258	s/ JT; s/ Agrupamento; c/ Rede
7	1	263	c/ JT; c/ Agrupamento; c/ Rede
8	2	325	c/ JT; s/ Agrupamento; c/ Rede

Quadro 11: Análise do tempo de processamento no *software TransCAD*.

Ordem de crescimento	Número do Problema	Tempo de processamento (s)	Características (fatores presentes e ausentes)
1	1	127	c/ JT; c/ Agrupamento; c/ Rede
2	3	129	c/ JT; c/ Agrupamento; s/ Rede
3	7	262	s/ JT; c/ Agrupamento; s/ Rede
4	5	291	s/ JT; c/ Agrupamento; c/ Rede
5	2	392	c/ JT; s/ Agrupamento; c/ Rede
6	4	403	c/ JT; s/ Agrupamento; s/ Rede
7	6	430	s/ JT; s/ Agrupamento; c/ Rede
8	8	525	s/ JT; s/ Agrupamento; s/ Rede

Analisando os quadros 10 e 11 percebe-se que os fatores aqui considerados, contribuem aumentando ou diminuindo o tempo de processamento das heurísticas. Nota-se também que esta contribuição não se dá de forma igual entre as duas heurísticas, ou seja, cada uma responde de forma diferente a presença ou ausência dos fatores. É difícil avaliar quantitativamente o quanto cada um destes fatores qualitativos interferem de forma estatisticamente significativa sobre o tempo de processamento das heurísticas, principalmente devido ao pequeno número de casos testados. Mesmo assim, tentou-se aferir o impacto destes fatores sobre o tempo de processamento de cada sistema através de regressão linear múltipla. A variável dependente deste modelo é representada pelo tempo de processamento (Y). As variáveis qualitativas independentes (janela de tempo - X1, rede viária - X2, e pré agrupamento - X3) foram introduzidas no modelo utilizando variáveis *dummy*.

$X1 = 0$, se não foi considerada restrição de janela de tempo;

$X1 = 1$, se foi considerada restrição de janela de tempo.

$X2 = 0$, se a roteirização não foi realizada sobre a rede viária;

$X2 = 1$, se foi realizada a roteirização sobre a rede viária.

$X3 = 0$, se os clientes não foram pré agrupados;

$X3 = 1$, se os clientes foram pré agrupados

Assim, para o *software Delivery*, tem-se: $Y = a + b X1 + c X2 + d X3$

Y	X1	X2	X3
19	0	0	0
20	0	0	1
23	1	0	0
26	1	0	1
254	0	1	1
258	0	1	0
263	1	1	1
325	1	1	0

Os quadros 12, 13 e 14 apresentam os resultados, para o *software Delivery*, da aplicação da técnica de regressão linear múltipla.

Quadro 12: Resumo dos resultados para o *software Delivery*

Estatística de regressão	
R múltiplo	0,99
R-Quadrado	0,98
R-quadrado ajustado	0,97
Erro padrão	22,36
Observações	8

Quadro 13: Coeficientes das variáveis e sua significância estatística (*software Delivery*)

	Coeficientes	Erro padrão	Stat t	valor-P	95% inferiores	95% superiores	Inferior 95,0%	Superior 95,0%
Interseção	19	15,81	1,20	0,30	-24,89	62,89	-24,89	62,89
Variável X 1	21,5	15,81	1,36	0,25	-22,39	65,39	-22,39	65,39
Variável X 2	253	15,81	16,01	0,00	209,11	296,89	209,11	296,89
Variável X 3	-15,5	15,81	-0,98	0,38	-59,39	28,39	-59,39	28,39

Quadro 14: Resíduos da regressão linear múltipla para o *software Delivery*.

Observação	Y previsto	Resíduos	Resíduos padrão
1	19	0	0,00
2	3,5	16,5	0,98
3	40,5	-17,5	-1,04
4	25	1	0,06
5	256,5	-2,5	-0,15
6	272	-14	-0,83
7	278	-15	-0,89
8	293,5	31,5	1,86

Dado o alto coeficiente de determinação (R múltiplo = 0,99), pode-se dizer que o modelo representa bem o comportamento do tempo de processamento da heurística do *software Delivery*. Através desta análise conclui-se que o tempo de processamento desta heurística é afetado, em primeiro lugar, pela rede viária, em segundo, pela janela de tempo e em terceiro lugar pelo pré agrupamento dos clientes, de forma que a rede viária e a janela de tempo provocam um aumento no tempo de processamento e o pré agrupamento provoca uma diminuição. Na realidade, o único fator que tem significância estatística (ver quadro 13) é a presença da rede viária. Os outros dois não são estatisticamente representativos.

No caso do *software TransCAD*, tem-se que:

Y	X1	X2	X3
127	1	1	1
129	1	0	1
262	0	0	1
291	0	1	1
392	1	1	0
403	1	0	0
430	0	1	0
525	0	0	0

Os quadros 15, 16 e 17 apresentam os resultados, para o *software TransCAD*, da aplicação da técnica de regressão linear múltipla.

Quadro 15: Resumo dos resultados para o *software TransCAD*

Estatística de regressão	
R múltiplo	0,98
R-Quadrado	0,95
R-quadrado ajustado	0,92
Erro padrão	40,50
Observações	8

Quadro 16: Coeficientes das variáveis e sua significância estatística (*software TransCAD*)

	Coeficientes	Erro padrão	Stat t	valor-P	95% inferiores	95% superiores	Inferior 95,0%	Superior 95,0%
Interseção	504,50	28,64	17,62	0,00	424,99	584,01	424,99	584,01
Variável X 1	-114,25	28,64	-3,99	0,02	-193,76	-34,74	-193,76	-34,74
Variável X 2	-19,75	28,64	-0,69	0,53	-99,26	59,76	-99,26	59,76
Variável X 3	-235,25	28,64	-8,21	0,00	-314,76	-155,74	-314,76	-155,74

Quadro 17: Resíduos da regressão linear múltipla para o *software Delivery*.

Observação	Y previsto	Resíduos	Resíduos padrão
1	135,25	-8,25	-0,27
2	155,00	-26,00	-0,85
3	269,25	-7,25	-0,24
4	249,50	41,50	1,36
5	370,50	21,50	0,70
6	390,25	12,75	0,42
7	484,75	-54,75	-1,79
8	504,50	20,50	0,67

Assim, o modelo representa bem o comportamento do *software TransCAD*, pois apresenta um alto coeficiente de determinação (R múltiplo = 0,98). Analisando o quadro 16, tem-se que o tempo de processamento da heurística do *software TransCAD* diminui quando os clientes são pré agrupados e, em menor grau, quando são consideradas restrições de janela de tempo nos clientes. Estatisticamente, a rede viária não afeta significativamente o tempo de processamento desta heurística.

É interessante como as heurísticas destes *software* possuem comportamentos opostos quando é analisado o tempo de processamento. Quanto maior a quantidade de restrições impostas durante o processamento no *software TransCAD*, menor o seu tempo de processamento. Por outro lado, no *software Delivery*, quanto maior o número de restrições levadas em consideração, maior será o seu tempo

computacional. As figuras abaixo ilustram o crescimento do tempo computacional em função dos fatores presentes e ausentes nos *software Delivery* e *TransCAD*, respectivamente.

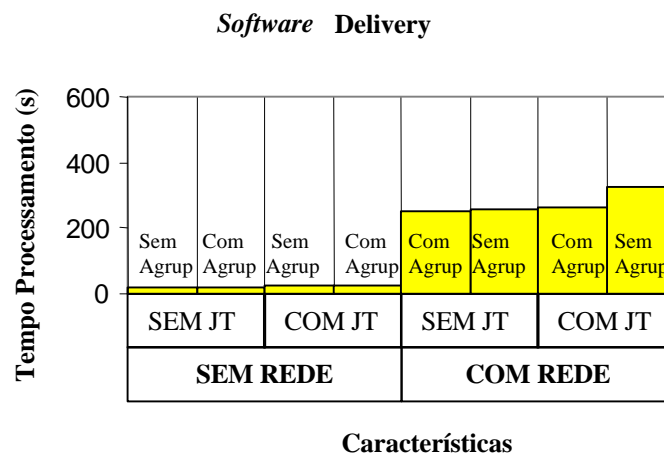


FIGURA 15 - REPRESENTAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO, NO *SOFTWARE DELIVERY*, EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES.

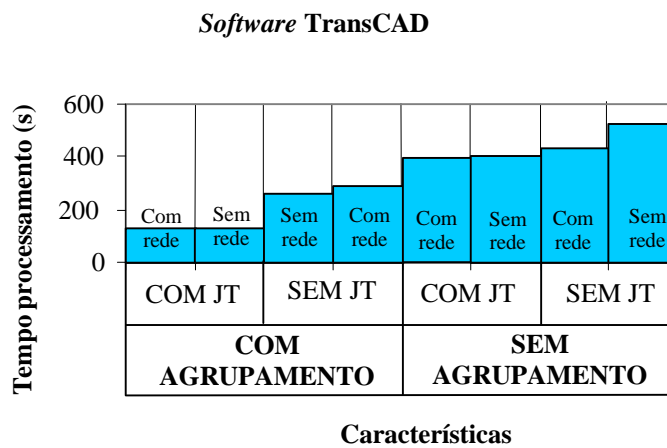


FIGURA 16- REPRESENTAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO, NO *SOFTWARE TRANS CAD*, EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES.

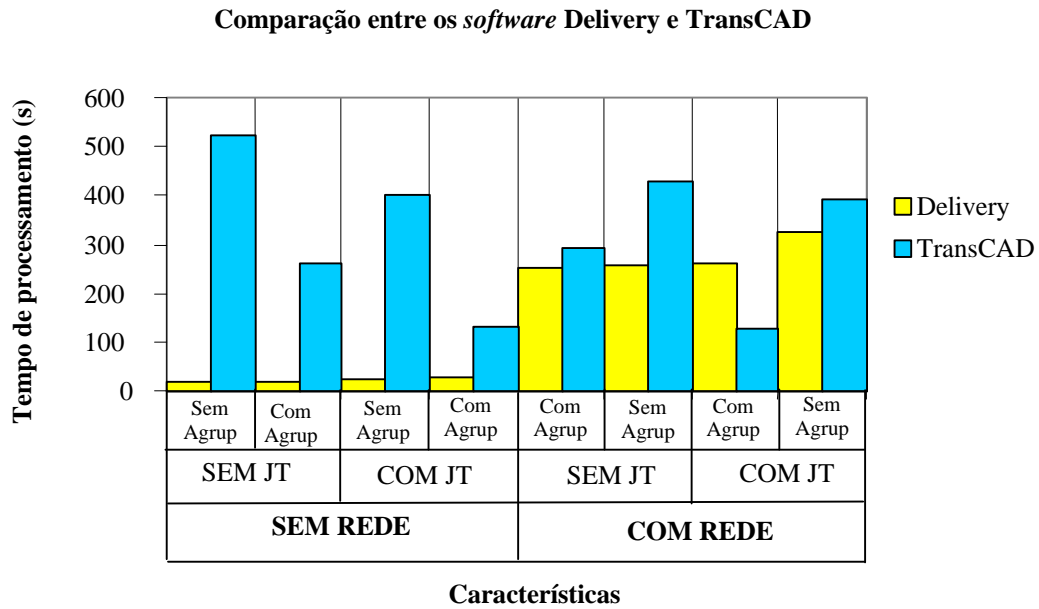


FIGURA 17. COMPARAÇÃO DO CRESCIMENTO DO TEMPO DE PROCESSAMENTO EM FUNÇÃO DA PRESENÇA E AUSÊNCIA DOS FATORES, NOS SOFTWARE *DELIVERY* E *TRANS CAD*.

8 CONCLUSÕES

Todos os objetivos propostos neste estudo foram atingidos. O objetivo principal, que consistiu em desenvolver uma metodologia para avaliação quantitativa do desempenho de um *software* comercial para roteirização de veículos, foi alcançado através do desenvolvimento dos objetivos secundários: 1) embasamento teórico e conceitual do problema de roteirização de veículos e análise de performance de heurísticas através de pesquisa bibliográfica; e 2) um estudo no cenário comercial para reconhecimento de várias características e fatores que podem interferir de forma positiva ou negativa em sistemas desta natureza .

A metodologia proposta foi conceitualmente estruturada sobre o método empírico comparativo, permitindo uma análise relativamente simples, de fácil reprodução, e bem documentada. Além disso, parece haver um forte consenso, por parte da comunidade acadêmica, na confiabilidade dos resultados desta análise.

Os resultados quantitativos do desempenho das heurísticas dos *software Delivery* e *TransCAD* sobre as classes de problemas de roteirização aqui testadas serão a seguir comentados. Também serão comentados alguns aspectos qualitativos desses sistemas, procurando apontar as vantagens e desvantagens de utilização das ferramentas.

No total, foram aplicados 51 problemas testes, com diferentes características visando analisar o desempenho das heurísticas sobre diferentes classes do problemas de roteirização de veículos. Destes 51 problemas, em apenas 3 deles o *software*

Delivery apresenta resultados melhores que o *software TransCAD* (ou seja, em aproximadamente 5% dos casos testados), sendo que os três pertenciam à mesma classe de problemas. Ainda assim, para esta mesma classe de problemas, a análise estatística concluiu, com nível de significância de 0,02, que a heurística do *software TransCAD* apresenta desempenho melhor que a heurística do *software Delivery*.

O pior comportamento apresentado pelo *software Delivery* corresponde ao tratamento de problemas em que os clientes possuem restrições de janela de tempo e estão geograficamente agrupados no espaço (conjunto de problemas C1 - Solomon, 1987). Com relação ao fator janela de tempo, é conveniente dizer aqui que este sistema oferece pouca flexibilidade ao usuário para o seu cadastramento, em decorrência da estrutura do banco de dados. Dada esta limitação, foi necessário aferir o desempenho desta heurística em relação ao *software TransCAD* e não em relação aos resultados encontrados na literatura. Assim, o *software Delivery* apresentou soluções com desvio médio 130% maior que as soluções apresentadas pelo *software TransCAD* no que diz respeito à distância percorrida, construindo um número médio de rotas 30,30% maior. É importante não perder o referencial de que o *software TransCAD* apresentou soluções com desvio médio de 20,60% na distância e 3,30% no número de rotas construída, acima da média dos melhores resultados encontrados na literatura revisada, para esta classe de problemas.

Também no caso real testado, o desempenho da heurística do *software Delivery* apresentou resultado ruim, onde a solução apresentada por este sistema perdeu para a solução conseguida por método intuitivo. Já o *software TransCAD* conseguiu uma redução, sobre a solução por método intuitivo, de 27,01% na distância total percorrida.

O *software TransCAD* demonstrou ser uma ferramenta bastante robusta apresentando, para todas as classes testadas, soluções que desviam aproximadamente em 20% para cima da média das melhores soluções encontradas na literatura revisada. Além disso, por se tratar de uma ferramenta SIG, é possível manipular, de forma rápida e precisa, dados de qualquer natureza. Assim, a principal vantagem

deste sistema é permitir uma formulação do problema teórico (modelo) bastante próxima do real. No entanto, dada a complexidade desta ferramenta, muitas dificuldades e problemas de ordem técnica e operacional podem ser encontrados durante sua manipulação, apesar da empresa que desenvolveu este sistema estar constantemente atualizando a versão e fornecer um serviço de suporte técnico eficiente e especializado. Pode-se afirmar ainda que a interface do sistema é amigável, embora as rotinas específicas exijam um certo tempo para o aprendizado.

Já o *software Delivery* não se mostrou uma ferramenta robusta, pois cada classe de problema testada apresentava soluções com desvio inesperado em relação à base de comparação. Agravando ainda mais este fato, dentro de uma mesma classe, foi possível encontrar soluções com resultados extremos (soluções muito boas e muito ruins). É conveniente mencionar a estrutura rígida deste sistema com relação à entrada de dados, o que limita muito a utilização da ferramenta. Muitas vezes, no decorrer deste trabalho, foi necessário utilizar o *software TransCAD* para manipular os dados dos problemas testes, de forma que se pudesse inserí-los como dados de entrada no *software Delivery*. Um exemplo disto, foi a construção da matriz distância origem_destino no *software TransCAD*, para que se pudesse aplicar os problemas testes no *software Delivery*. Como as localizações dos clientes nesses problemas são especificadas em função de coordenadas (x,y), não seria possível aplicar tais testes no *software Delivery*, uma vez que o sistema não reconhece este tipo de dados, não fosse a preparação da matriz origem_destino com o auxílio de uma outra ferramenta. Ainda assim, a principal vantagem deste sistema diz respeito a sua interface, bastante simples e amigável. Este sistema permite ainda que o problema seja formulado considerando uma grande variedade de restrições. Alterações manuais das soluções podem ser realizadas por parte do usuário, apenas com a utilização do *mouse*, de forma que o sistema providencia rapidamente as soluções alteradas.

Com relação ao tempo de processamento, os dois *software* apresentaram comportamentos bastante distintos. Apesar do pequeno número de casos analisados, pode-se ordenar os fatores que contribuem aumentando ou diminuindo, o tempos de processamento das heurísticas nos *software* testados. O tempo de processamento no

software TransCAD diminui quando são consideradas restrições de pré agrupamento dos clientes por veículo e, em menor grau, quando considera-se restrições de janelas de tempo nos clientes. A presença de uma rede viária não interfere significativamente no seu tempo de processamento. Já o *software Delivery* apresenta um aumento considerável no tempo de processamento quando é utilizada uma rede viária. A presença de restrições de janelas de tempo nos clientes também contribui, de forma mais suave, aumentando o tempo, enquanto o pré agrupamento dos clientes contribui diminuindo, apesar destes dois últimos fatores não demonstrarem interferir significativamente quando analisados estatisticamente.

É importante deixar claro que a empresa que desenvolveu o *software Delivery* foi notificada dos resultados apresentado pelo seu sistema. Como uma preocupação em garantir a integridade do seu produto, alterações e melhorias estão sendo implementadas. Futuras avaliações podem perfeitamente fazer uso da metodologia aqui proposta que, neste sentido, se torna uma referência não só para os fabricantes de *software*, mas principalmente para os usuários.

9 REFERÊNCIAS BIBLIOGRÁFICAS

- ALTIKEMER, K; GAVISH, B. (1991). Parallel savings based heuristics for the Delivery problem. *Operational Research*, n.39, p. 456-469.
- ASSAD, A. A.; GOLDEN, L. B. (1984). A decision-theoretic framework for comparing heuristics. *European Journal of Operatinal Research*, v.18, p. 167-171.
- ASSAD, A. A.; GOLDEN, L. B.; WASIL, E. A.; BAKER, E. (1986). Experimentation in optimization. *European Journal of Operational Research*, v.27, p.1-16.
- ASSAD, A. A. (1988). Modeling and implementation issues in vehicle routing. In: GOLDEN, L. B.; ASSAD, A. A. (eds). *Vehicle Routing: Methods and Studies*. North Holland, Amsterdam, p. 7-46.
- BALAS, E.; CHRISTOFIDES, N. (1981). A restricted Lagrangean approach to the traveling salesman problem. *Mathematical Programming*, n.21, p. 19-46.
- BALL, M.; MAGAZINE, M. (1981). The design and analysis of heuristics. *Networks*, n.11, p. 215-219.
- BALLOU, R. H. (1993). *Logística Empresarial: transporte, administração de materiais e distribuição física*. Tradução Hugo T. Y. – São Paulo: Atlas.
- BODIN, L.; BERMAN, L. (1979). Routing and scheduling of school buses by computer. *Transportation Science*, n.13, p. 113-129.
- BODIN, L. D.; GOLDEN, B.; ASSAD, A.; BALL, M. (1983). Routing and scheduling of vehicle and crews: The state of the art. *Computers and Operational Research*, v.9, p. 63-212.
- BODIN, L.; KURSH, S. (1978). A computer-assisted system for the routing and scheduling of street sweepers. *Operational Research*, v.26, n.4, 525-537.

- BODIN, L.; KURSH, S. (1979). A detailed description of a street sweeper routing and scheduling system. *Computers & Operations Research*, n.14B, p. 115-120.
- BARBAROSOGLU, G.; OZGUR, D. (1999). A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, v. 26, p.255-270.
- BEASLEY, J. E. (1983). Route first-cluster second methods for vehicle routing. *Omega* v.118, pp.403-408.
- BEASLEY, J.E. (1990). OR Library: distributing test problems by electronic mail. *Journal of Operational Research Society*, 41, 1069-1072.
- BLAND, R.; CHERIYAN, J.; JENSEN, D.; LADÁNYI, L. (1993). An empirical study of min cost flow algorithms. In: JOHNSON, D. & MCGEOCH, C. (Eds.), *Network Flow and Matching: First DIMACS Implementation Challenge*, Vol. 12 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 119-156. American Mathematical Society, Providence, RI.
- CHRISTOFIDES, N. (1981). Uses of a vehicle routing and scheduling system in strategic distribution planning. *Scandinavian Journal of Mat Admin*, v.7, n.2, p.39-55.
- CHRISTOFIDES, N (1985). Vehicle Routing. In:LAWLER, E. L.; LENSTRA, J. K.; KAN, A. H. G. R.; SHMOYS, D. B (ed). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons.
- CHRISTOFIDES, N.; EILON, S. (1969). An algorithm for the vehicle Dispatching problem. *Operational Research*, n. 20, p. 309-318.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. (1978). The vehicle routing problem. *Urbino Working Paper*, July, 1978.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P.; SANDI, C. (1979). *Combinatorial optimization*. John Wiley, Chichester.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxation. *Mathematical Programming*, n.20, p. 255-282
- CLARKE, G.; WRIGHT, J. (1964). Scheduling of vehicles from a central depot to a number of Delivery points. *Operational Research*, n.12, p. 568-581.
- CUNHA, C. B. (1997). *Uma contribuição para o problema de roteirização de veículos com restrições operacionais*. São Paulo. Tese (Doutorado). Escola Politécnica Universidade São Paulo, Departamento de Engenharia de Transportes, Universidade de São Paulo.

- DESROCHERS, M.; LENSTRA, J. K.; SAVELSBERGH, M. W. P. (1990). A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, v.46, p.322-332.
- DESROCHERS, M.; VERHOOG, T. W. (1989). *A Matching Based Savings Algorithm for the vehicle routing problem*. Cahier du GERAD G-89-04. École des Hautes Études Commerciales de Montréal.
- DESROCHERS, M.; VERHOOG, T. W. (1990). A new heuristic for the fleet size and mix vehicle routing problem. *Computers & Operational Research*, v.18, p.263-274.
- EILON, S.; EATSON-GANDY, C.; CHRISTOFIDES, N. (1971). *Distribution Management: Mathematical Modeling and Practical Analysis*. Hafner, New York.
- FISHER, M. L.; JAIKUMAR, R. (1981). *A generalized assignment heuristic for vehicle routing*. *Networks*, v. 11, pp. 109-124.
- GAREY, M.; JOHNSON, D. (1979). *Computers and Intractability: A guide to the Theory of NP-Completeness*. Freeman, San Francisco.
- GARCIA, B. L.; POTVIN, J. Y.; ROUSSEAU, J. M. (1994). A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research*, v.21, n.9, p.1025-1033.
- GASKELL, T. J. (1967). Bases for vehicle fleet scheduling. *Operational Research*, n.18, p.281-295.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, v. 40, n.10, p.1276-1290.
- GILLET, A.; JOHNSON, J. (1976). Multi-terminal vehicle dispatch algorithm. *Omega*, n.4, p.711-718
- GILLET, B.; MILLER, L (1974). A heuristic algorithm for the vehicle dispatch problem. *Operational Research*, n.22, p.340-349.
- GOLDEN, B.; ASSAD, A.; LEVY, L.; GHEYSENS, F. (1982). *The fleet size and mix vehicle routing problem*. Management Science & Statistics Working Paper, (Report n.82-020).
- GOLDEN, B. L.; STEWART, W. R. (1985). Empirical analysis of heuristics. In LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G.; SHMOYS, D. B. (eds.). *The Traveling Salesman Problem*. John Wiley & Sons. Ltd. New York, NY.

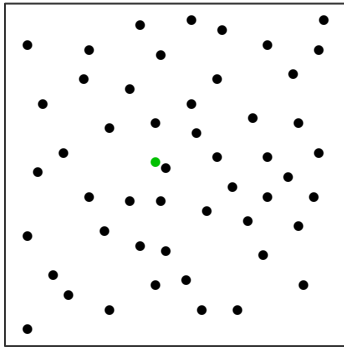
- GOLDEN, B. L.; WASIL, E. A.; BAKER, E. (1986). Experimentation in optimization. *European Journal of Operational Research*, n. 27, pp. 1-16.
- GOLDEN, B.; MAGNANTI, T.; NGUYEN, H. (1977). Implementing vehicle routing algorithms. *Networks*, n.7, p.113-148.
- GOLDEN, B.; WONG, R. (1981). Capacitated arc routing problems. *Networks*, n.11, v.3, p.305-315.
- HAIMOVICH, M.; RINNOOY KAN, A. H. G. (1985). Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, v.10, p.527-542.
- HAYS, R. (1967). *The Delivery problem*. Department of Management Science, Carnegie Institute of technology, Pittsburgh, PA. [12:1] (Report 106).
- HOOKER, J. (1994). Needed: An empirical science of algorithms. *Operations Research*, 42 (2), 201-212.
- JOHNSON, D. S.; PAPADIMITRIOU, C. H. (1985). Performance guarantees for heuristics. In: LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G.; SHMOYS, D. B. (eds.). *The Traveling Salesman Problem*. John Wiley & Sons. Ltd. New York, NY.
- KARP, R. (1977). Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. *Mathematics of Operations Research*, n.2, p. 209-224.
- KELLY, J. P.; XU, J. (1999). A set-partitioning-based heuristic for the vehicle routing problem. *INFORMS Journal on Computing*, v. 11, n.2, p.161-172.
- LAPORTE, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, v. 59, n.3, p. 345-358.
- LAWLER, E. L.; LENSTRA, J. K.; KAN, A. H. G. R.; SHMOYS, D. B. (1985). *The traveling salesman problem. A guide tour of combinatorial optimization*. A Wiley-Interscience Publication. Chichester, New York, Brisbane, Toronto, Singapore: John Wiley & Sons.
- LENSTRA, J. K.; RINNOOY KAN, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, n.11, pp.221-227.
- LIN, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Computer Journal*, n.44, p. 2245-2269.
- LIN, S.; KERNIGHAN, B. W. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operational Research*, n.27, p. 503-511.

- LIU, F. F.; SHEN, S. (1999). A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. *European Journal of Operational Research*, v. 118, n. 3, p.485-504.
- MAGNANTI, T. (1981). Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects. *Networks*, v.11, n.2, p. 179-214.
- McGEOCH, C. (1995). Toward an experimental method for algorithm simulation. *ORSA Journal on Computing*, to appear.
- MOLE, R. H.; JAMESON, S. R. (1976). A sequential route-building algorithm employing a generalised savings criterion. *Operational Research*, v. 27, pp. 503-511.
- NELSON, M.; NYGARD, K.; GRIFFIN, J.; SHREVE, W. (1985). Implementation techniques for the vehicle routing problem. *Computers & Operational Research*, v.12, p. 273-283.
- NEWTON, R.; THOMAS, W. (1969). Design of school bus routes by computer. *Socio-Economic Planning Science*, n.3, p.75-85.
- OSMAN, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41, pp. 421-451.
- PAESSENS, H. (1988). Saving algorithms for the vehicle routing. *European Journal of Operational Research*, v.34, pp. 336-344.
- PELIZARO, C; PEREIRA, M. A.; SILVA, A. N. R. (1998). Uma solução apoiada em SIG para o problema de roteamento de veículos em uma cidade média. *Revista Universidade de Guarulhos - Ciências Exatas e Tecnológicas*, Guarulhos, v.3, n.4, pp. 13-18, ago.
- PUREZA, V. M.; FRANÇA, P. M. (1991). *Vehicle routing problems via tabu search metaheuristic*. Montreal. Centre de recherche su les transports (Publication CRT-747).
- READ, W. F.; MULLER, M. S. (1991). The state of quality in logistics. *International Journal of Physical Distribution & Logistics Management*, v.21, n.6, p. 32-47.
- REEVES, C. R. (1993). *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons. Inc. New York, NY.
- RESENDE M. G. C.; STEWART W. R.; KELLY J. P.; GOLDEN, B. L.; BARR, R. S. (1995). Designing and Reporting on Computational Experiments with Heuristic Methods. *Working Paper*. June,16, 1995.

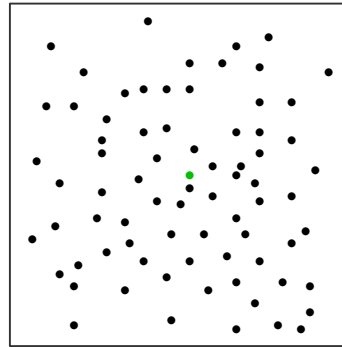
- ROUSSEAU, J. M. (1988). *Customization versus a general purpose code for routing and scheduling problems: a point of view*. In: GOLDEN, L. B.; ASSAD, A. A. (eds). *Modeling and implementation issues in vehicle routing: Methods and Studies*. North Holland, Amsterdam, p. 469-479.
- RUSSELL, R. A. (1977). An effective heuristic for the M-tour traveling salesman problem with some side conditions. *Operational Research*, v. 25, p. 517-524.
- RUSSELL, R. (1995). Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science*, v.29, n. 2, p.156-166.
- SEMET, F.; TAILLARD, E. (1993). Solving real life vehicle routing problems efficiently using taboo search. *Annals of Operational Research*, n.41, pp.469-488.
- SILVA, A. N. R.; LOTTI, C. P.; MARGARIDO, S. A; PAMPOLHA, V. M. P. (1996). Quem não tem Tiger... Fator GIS, Curitiba, 4 (14): 18-19, jun./jul.
- SOLOMON, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time windows constraints. *Operations Research*, v. 35, n.2, pp. 254-265.
- SOLOMON, M. M.; DESROSIERS, J. (1988). Time window constrained routing and scheduling problems. *Transportation Science*, v. 22, n. 1, pp.1-13.
- SWAIT, J. D. (1990). *Implementação de algoritmos de pesquisa operacional, Volume I: Fundamentos Computacionais – algoritmos e estruturas de dados*. Pré-edição do livro do autor.
- TAILLARD, E. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, v. 23, p. 661-673.
- YELLOW, P. (1970). A computation modification to the savings method of vehicle scheduling. *Operational Research*, n.21, p. 281-283.

APÊNDICE - ILUSTRAÇÃO DA DISPERSÃO GEOGRÁFICA DOS CLIENTES E DEPÓSITO NOS PROBLEMA TESTES.

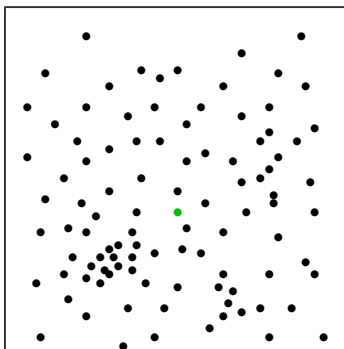
1. CONJUNTO DE PROBLEMAS TESTE DE CHRISTOFODES *et al.* (1979)



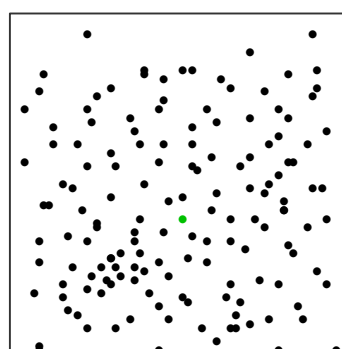
VRPNC1 e 6



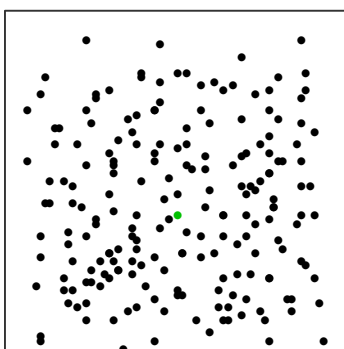
VRPNC2 e 7



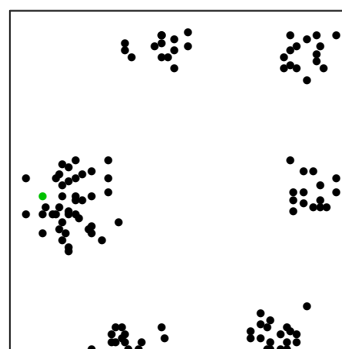
VRPNC3 e 8



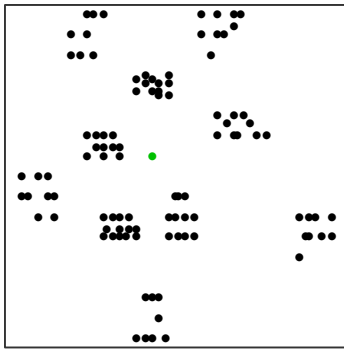
VRPNC4 e 9



VRPNC5 e 10

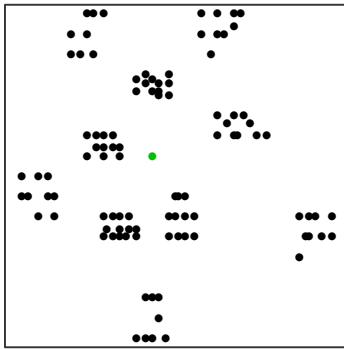


VRPNC11 e 13

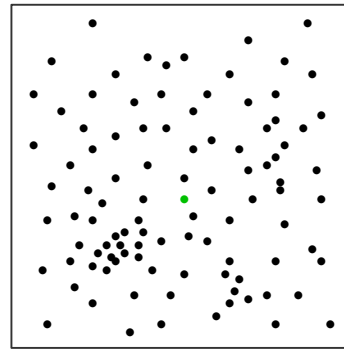


VRPNC 12 e 14

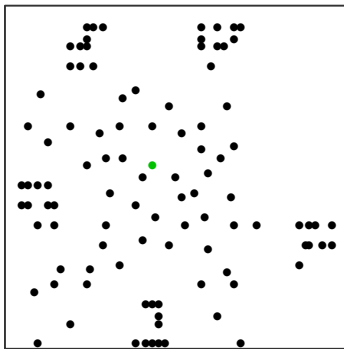
2. CONJUNTO DE PROBLEMAS TESTE DE SOLOMON (1987)



C1

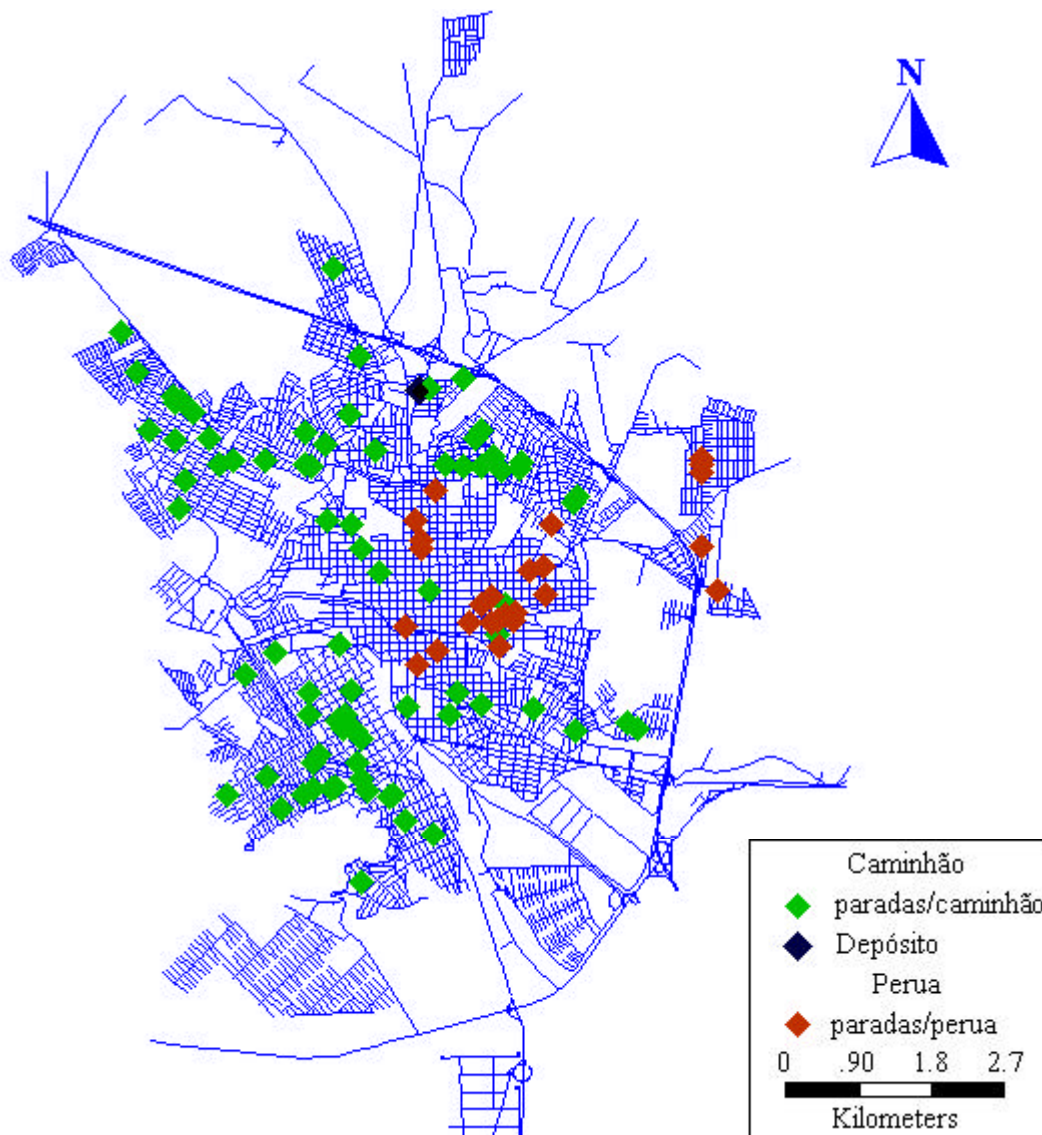


R1



RC1

3. PROBLEMA COM DADOS REAIS



Mapa da cidade de São Carlos com a localização dos clientes a serem atendidos no estudo de caso real