

UNIVERSIDADE FEDERAL FLUMINENSE

**ALGORITMOS PARA PROBLEMAS REAIS DE ROTEAMENTO DE
VEÍCULOS: UMA ANÁLISE COMPARATIVA**

FILIPPE QUINTIERI LIMA

NITERÓI

2005

Algoritmos para Problemas Reais de Roteamento de Veículos:
Uma Análise Comparativa

Filipe Quintieri Lima

Dissertação apresentada ao Curso de Pós-Graduação
em Ciência da Computação da Universidade Federal
Fluminense como parte dos requisitos necessários para
obtenção do título de Mestre em Ciência da Computação

Orientadores: Lúcia Maria de Assumpção Drummond
Eduardo Uchoa Barboza

Niterói
2005

Algoritmos para Problemas Reais de Roteamento de Veículos:
Uma Análise Comparativa

Filipe Quintieri Lima

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação

Lúcia Maria de Assumpção Drummond
Instituto de Computação – UFF

Eduardo Uchoa Barboza
Dept. Engenharia de Produção – UFF

Marco Antônio Farah Caldas
Dept. Engenharia de Produção – UFF

Marcus Vinicius Soledade Poggi de Aragão
PUC – Rio

Adriana Cesário de Faria Alvim
ITA – SP

Niterói
2005

Este trabalho é dedicado a toda minha família, sempre presente nos momentos mais difíceis da minha vida.

À minha avó materna, Maria Iza Ferraz Quintieri, minha maior incentivadora.

Aos meus pais, Maria Vitória Quintieri Lima e Jorge Siqueira de Lima, sempre presentes na minha vida, com seu amor, carinho e apóio.

À minha tia Izabel Cristina Quintieri da Silva pelo apóio e incentivo no momento mais difícil desta caminhada.

Agradecimentos

Meus agradecimentos vão a todas as pessoas que direta ou indiretamente participaram deste trabalho e conviveram neste período comigo, desde os momentos mais alegres aos mais difíceis de mais esta etapa de minha vida.

Aos meus orientadores Eduardo Uchoa e Lúcia Drummond que me apresentaram todas as possibilidades e ferramentas para o desenvolvimento deste trabalho. O apoio, amizade e credibilidade dos meus orientadores foram fundamentais para a conclusão deste trabalho. Eduardo e Lúcia saibam que participaram de um dos momentos mais importantes de minha vida e foram de fundamental importância para meu crescimento profissional e principalmente pessoal, serei eternamente grato pela dedicação e compreensão.

Aos meus amigos de mestrado, companheiros desta luta, muitos de cidades distantes, ou até de outros estados, que em muitos momentos foram os amigos necessários para que eu pudesse prosseguir atrás desta conquista. Fica aqui meu agradecimento principalmente para duas pessoas que não sairão da minha memória, Euller Marinho e Haroldo Santos, passamos muitos momentos juntos em diversas disciplinas, a amizade dos dois será eterna.

Aos meus companheiros de apartamento, Canella, Pereira, Jean, Fábio, Zé, Nei e Igor, muitos momentos engraçados e críticos passamos juntos, espero que todos tenham sucesso em suas futuras vidas profissionais e pessoais, que as experiências vivenciadas em nossa república sirvam para todos como lições para nossas vidas.

Aos meus colegas de trabalho do SENAI de Nova Friburgo pelos bons conselhos e apoio nos muitos momentos difíceis. Especialmente a Oswaldo, pelo seu iBook, utilíssimo na reta final da dissertação.

Às minhas chefes, Catarina, Claudia, Lúcia e Thereza, pela compreensão e apoio, principalmente nos momentos em que estive ausente no SENAI.

Resumo

Este trabalho apresenta uma comparação entre diversos algoritmos e abordagens para problemas de roteamento de veículos típicos de serem encontrados na prática. Essa comparação foi realizada sobre um grupo de instâncias criadas a partir de dados reais de uma empresa varejista atuante na cidade de São Paulo. Este trabalho propõe três heurísticas construtivas e cinco algoritmos de busca local para aperfeiçoamento das soluções obtidas por essas heurísticas ou pelas heurísticas contidas em um pacote comercial de roteamento. O trabalho também apresenta duas implementações de metaheurísticas a partir da técnica GRASP. É feita uma análise comparativa entre os resultados obtidos a partir de todas essas possíveis abordagens e os resultados obtidos a partir de um pacote comercial de roteirização. Os resultados obtidos trazem à discussão a importância econômica da utilização de algoritmos mais sofisticados para roteamento de veículos nas empresas, possivelmente em adição àqueles presentes em pacotes comerciais.

Palavras-chaves: Heurísticas, Metaheurísticas, Buscas Locais, Logística de Distribuição, Roteamento.

Abstract

This work presents a comparison of algorithms and approaches for vehicle routing problems like those found in typical real world applications. This comparison was performed over a set of instances created from real data provided by a large retailer in the city of São Paulo. This work proposes three constructive heuristic and five local search algorithms. The latter can also be used to improve the solutions provided by a commercial routing software package. The work also presents two different implementations of a GRASP metaheuristic. A comparative analysis of all such approaches is performed. The obtained results stress the economic importance of applying more sophisticated routing algorithms to real world problems, possibly in addition to those currently available in the commercial packages.

Keywords: Heuristics, Metaheuristics, Local Search, Logistics, Routing.

Sumário

1. Introdução	11
2. Problema de Roteamento de Veículos	13
2.1. Extensões	15
2.2. Caso em Estudo	17
3. PACOTE ROTEAMENTO DE VEÍCULOS.....	20
4. Heurísticas Construtivas	26
4.1. Vizinho mais Próximo	26
4.2. Economias	30
4.2.1. Versão 1	32
4.2.2. Versão 2	33
5. Buscas Locais.....	35
5.1. Eliminar Rotas	35
5.2. Mudar Atendimentos.....	37
5.3. Trocar Atendimentos	40
5.4. Resequenciar Atendimentos.....	42
5.5. Inserir Atendimentos.....	43
6. Metaheurística GRASP.....	45
6.1. Versão Vizinho mais Próximo	47
6.2. Versão Economias 2.....	48
7. Instâncias.....	50
8. Resultados Computacionais	52
9. Conclusões e Propostas Futuras.....	58
Referências Bibliográficas	60
Apêndice A – Estrutura dos dados nas instâncias	63
Apêndice B – Resultados individual para cada instância	71

Lista de Figuras

Figura 1: Exemplo de uma solução para uma instância do PRV	15
Figura 2: Processo de criação de rotas no pacote	24
Figura 3: Representação dos clientes (demandas), depósito e custos de distância.....	27
Figura 4: Veículo inicia rota saindo do depósito para o cliente viável mais próximo.....	27
Figura 5: Visita ao segundo cliente viável mais próximo, a partir da primeira visita.....	28
Figura 6: Não existindo mais atendimentos viáveis, o veículo retorna ao depósito	28
Figura 7: Solução final após a execução da lógica do Vizinho mais Próximo.....	29
Figura 8: Pseudocódigo Heurística Vizinho mais Próximo	29
Figura 9: Representação dos clientes (demandas), depósito e custos de distância.....	30
Figura 10: Conjunto de rotas inicialmente criadas com a lista de economias gerada.....	31
Figura 11: União das rotas, a partir da economia gerada com os clientes 3 e 4.....	31
Figura 12: Solução final após a execução da lógica da heurística de Economias	32
Figura 13: Pseudocódigo Heurística de Economias – versão 1	33
Figura 14: Pseudocódigo Heurística Economias – versão 2.....	34
Figura 15: Pseudocódigo Busca Local Eliminar Rotas.....	37
Figura 16: Pseudocódigo Busca Local Mudar Atendimentos	39
Figura 17: Pseudocódigo Busca Local Trocar Atendimentos.....	41
Figura 18: Pseudocódigo Busca Local Resequenciar Atendimentos	43
Figura 19: Pseudocódigo Busca Local Inserir Atendimentos	44
Figura 20: GRASP para o Problema Roteamento de Veículos	46
Figura 21: Pseudocódigo Construtivo Vizinho mais Próximo	48
Figura 22: Pseudocódigo Economias – versão 2	49
Figura 23: Visualização gráfica das comparação entre abordagens.....	57

Lista de Tabelas

Tabela 1: Descrição dos dados referentes a Janela de Tempo	18
Tabela 2: Descrição dos dados referentes a Veículos	18
Tabela 3: Descrição dos dados referentes a Docas.....	19
Tabela 4: Relação do número de registros para as instâncias.....	51
Tabela 5: Classificação média dos algoritmos nas instâncias.....	53
Tabela 6: Resultados da heurística Inserir Atendimentos	55
Tabela 7: Visualização tabular das comparação entre abordagens.....	56
Tabela 8: Estrutura do arquivo de Locais	63
Tabela 9: Estrutura do arquivo de Janelas de Tempo.....	64
Tabela 10: Estrutura do arquivo de Produtos	65
Tabela 11: Estrutura do Arquivo de Pedidos	66
Tabela 12: Estrutura do arquivo de Itens de Pedido	67
Tabela 13: Estrutura do arquivo de Veículos.....	67
Tabela 14: Resultados instância 065a.....	72
Tabela 15: Resultados instância 065b.....	74
Tabela 16: Resultados instância 080a.....	76
Tabela 17: Resultados instância 080b.....	78
Tabela 18: Resultados instância 095a.....	80
Tabela 19: Resultados instância 095b.....	82
Tabela 20: Resultados instância 110a.....	84
Tabela 21: Resultados instância 110b.....	86
Tabela 22: Resultados instância 125a.....	88
Tabela 23: Resultados instância 125b.....	90
Tabela 24: Resultados instância 140a.....	92
Tabela 25: Resultados instância 140b.....	94
Tabela 26: Resultados instância 155a.....	96
Tabela 27: Resultados instância 155b.....	98
Tabela 28: Resultados instância 170a.....	100
Tabela 29: Resultados instância 170b.....	102
Tabela 30: Resultados instância 185a.....	104
Tabela 31: Resultados instância 185b.....	106
Tabela 32: Resultados instância 200a.....	108
Tabela 33: Resultados instância 200b.....	110

1. Introdução

A etapa de transporte e distribuição de produtos representa grande parte dos custos logísticos de uma empresa. Um sistema de transporte e distribuição eficiente e com custos otimizados contribui para reduzir os preços das mercadorias, com conseqüente melhoria de competitividade. O Problema de Roteamento de Veículos (PRV) consiste na escolha de rotas de veículos que minimizem custos e tempo a fim de atender as demandas de um conjunto de clientes. Esse problema, também conhecido como Problema de Escalonamento de Veículos, é um dos mais estudados na área de otimização combinatória.

A grande maioria das centenas de estudos acadêmicos já apresentados para o Problema de Roteamento de Veículos trata de algoritmos, exatos ou heurísticos, para as suas variantes mais simples. Apesar dos bons resultados obtidos, esses algoritmos raramente são utilizados em situações reais, pois essas variantes simples consideram apenas uma pequena parte das restrições e características dos problemas que costumam ocorrer numa típica empresa. Na prática, normalmente utilizam-se pacotes comerciais que têm uma grande flexibilidade, capazes de se adaptar às condições particulares de uma empresa através de uma extensa parametrização. Por outro lado, a grande generalidade oferecida, por estes pacotes, tem como contrapartida o uso de algoritmos heurísticos mais simples, que muitas vezes não são capazes de fornecer soluções de boa qualidade.

O objetivo deste trabalho é apresentar algoritmos adequados ao conjunto de restrições mais comuns nos problemas de roteamento reais. Esses algoritmos devem ser capazes de obter soluções melhores do que as fornecidas por um pacote comercial utilizado no mercado, possivelmente partindo das próprias soluções geradas por esse pacote. Foram desenvolvidas três heurísticas construtivas, cinco algoritmos de busca local e duas heurísticas baseadas na metaheurísticas GRASP.

As análises e comparações entre todos os métodos apresentados foram realizadas sobre um conjunto de instâncias criadas a partir de dados reais, obtidos do problema de entregas de uma loja de departamentos da cidade de São Paulo. O estudo traz à discussão a possibilidade de melhorias significativas na qualidade das soluções obtidas pelos pacotes de roteamento atualmente utilizados em muitas empresas.

Este trabalho segue organizado da seguinte maneira. O presente capítulo introduz o tema da dissertação e a sua motivação. No Capítulo 2, é realizada uma descrição do Problema de Roteamento de Veículos em suas principais variantes, além de se apresentar um modelo matemático para a variante a ser estudada. Uma introdução ao

processo de funcionamento de um pacote de roteamento e seus respectivos métodos heurísticos está presente no Capítulo 3. As três heurísticas construtivas usadas neste trabalho são apresentadas no Capítulo 4. O Capítulo 5 trata de buscas locais, propondo cinco algoritmos para o aprimoramento das soluções obtidas através dos algoritmos construtivos. Duas versões heurísticas baseadas na metaheurísticas GRASP são descritas no Capítulo 6. Os resultados computacionais dos algoritmos propostos são analisados e comparados, durante o Capítulo 7, juntamente com os resultados obtidos através dos métodos disponíveis no pacote de roteamento utilizada nas comparações. No Capítulo 8, a dissertação é concluída com algumas idéias de futuros trabalhos.

2. Problema de Roteamento de Veículos

O Problema Roteamento de Veículos (PRV), enunciado inicialmente em [3], tem como dados um conjunto de depósitos e clientes dispersos, um conjunto de demandas de transporte e uma frota de veículos. O objetivo é criar rotas para que os veículos possam transportar as demandas, respeitando uma série de restrições operacionais e minimizando os custos envolvidos, de forma a não violar as restrições impostas.

Esse tipo de problema aparece em um grande número de situações. Pode-se citar atividades como entrega ou recolhimento de mercadorias, transporte de pessoas, coleta de lixo, entrega postal, entre outras. Esses exemplos mostram a importância do estudo dessa classe de problemas do ponto de vista do econômico.

Existem diversas variantes do PRV, cada uma adequada a uma situação em que esse tipo de problema ocorre. Praticamente todas as variantes do PRV pertencem ao grupo de problemas NP-difíceis, visto que esses problemas são uma generalização do clássico Problema do Caixeiro Viajante (PCV). A resolução exata desses problemas provavelmente exige um esforço computacional que cresce exponencialmente com o número de clientes a serem atendidos.

Uma formulação matemática para uma variante básica do PRV é descrita em [20] e mostrada a seguir.

Seja $I = (G, C, q, Q)$ uma instância, onde se tem um grafo direcionado $G = (V, E)$ ($|V| = N$, $V = \{1, \dots, N\}$, nó 1 é o depósito), uma matriz $N \times N$ de custos $C = (c_{ij})$ (cada arco (i, j) possui um custo c_{ij}), uma lista de tamanho N com q demandas (cada nó i possui uma demanda de q_i , sendo $q_1=0$) e uma lista de tamanho M com Q capacidades (cada veículo possui capacidade Q_k $1 \leq k \leq M$). Os elementos em V diferentes do depósito recebem o nome de clientes. Neste modelo básico do PRV existem restrições de capacidade, a frota é heterogênea e existe um único depósito. Seja X_{ijk} uma variável binária igual a 1, se o veículo k visita o nó j imediatamente após o nó i , e 0, caso contrário. Os índices da formulação representam:

i = origem;

j = destino;

k = veículo;

N = total de demandas;

M = total de veículos;

C = custos.

Minimizar:

$$\sum_{i,j=1}^N c_{ij} \sum_{k=1}^M x_{ijk}$$

Sujeito a:

$$\sum_{k=1}^M \sum_{j=1}^N x_{ijk} = \begin{cases} 1, & 2 \leq i \leq N \\ M, & i = 1; \end{cases} \quad 1$$

$$\sum_{i,j=1}^N q_i x_{ijk} \leq Q_k, \quad 1 \leq k \leq M; \quad 2$$

$$\sum_{j=1}^N x_{ijk} = \sum_{j=1}^N x_{jik}, \quad 1 \leq i \leq N, 1 \leq k \leq M; \quad 3$$

$$\sum_{k=1}^M \sum_{(i,j) \in S} x_{ijk} \leq |S| - 1, \quad \text{para todos } S \in 2^{V \setminus \{1\}}; \quad 4$$

$$x_{ijk} \in \{0,1\}, \quad 1 \leq i, j \leq N, 1 \leq k \leq M. \quad 5$$

A restrição 1 assegura que somente um veículo pode sair de um cliente, exceto pelo depósito, onde M veículos podem sair a partir do mesmo. Restrição 2 leva em consideração a capacidade de cada veículo. Restrição 3 diz que se um veículo chega em um cliente ele deve sair desse cliente. A restrição 4 previne sub-rotas, prevenindo que uma rota de um determinado veículo não forme múltiplas rotas desconectadas. A última restrição, 5, delimita os possíveis valores que a variável X possa assumir, tornando-a binária.

Na Figura 1 é mostrado um exemplo de solução para uma instância dessa variante do PRV que consiste em: um conjunto de cinco clientes (2, 3, 4, 5 e 6), onde a quantidade demandada por cada cliente está referenciada entre colchetes, e três rotas com diferentes veículos, A, B e C respectivamente para cada rota. O veículo da rota A possui capacidade de 10, B de 5 e C de 8. Todos partem de um único depósito, 1, e retornam para o mesmo ao final da rota.

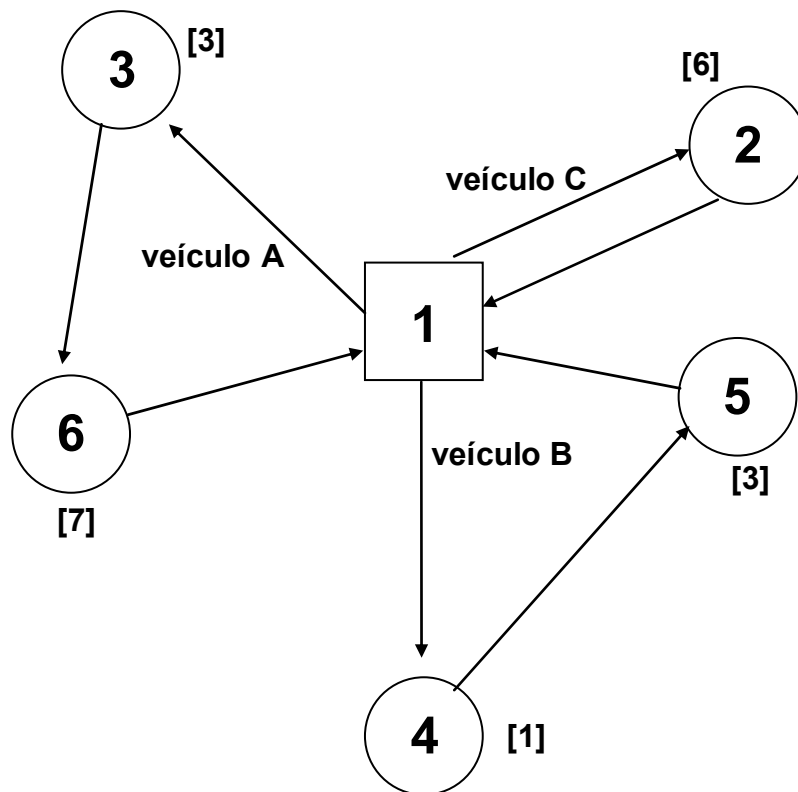


Figura 1: Exemplo de uma solução para uma instância do PRV

Existe uma considerável quantidade de trabalhos acadêmicos que tratam do Problema de Roteamento de Veículos. Uma classificação das principais variantes encontradas para esta classe de problemas encontra-se em [06]. Duas importantes relações bibliográficas encontram-se em [14] e [15]. Dois livros que reúnem trabalhos sobre o tema são [16], [17] e coletâneas sobre o assunto podem ser encontradas em [20] e [21]. Entre as referências centradas nos métodos para solucionar este tipo de problema podemos citar: programação inteira [07], [11] e [12], heurísticas [05], [08], [09], [10] e [19], e metaheurísticas [13] e [18].

2.1. Extensões

Para cada variante ou extensão proposta do Problema de Roteamento de Veículos existem vários estudos realizados no meio acadêmico, porém é muito comum se encontrar nomes diferentes para problemas iguais. As novas extensões, que surgem ao decorrer dos anos, possuem o intuito de melhorar e aprimorar os modelos baseados no PRV, para cada vez mais aproximar-se das aplicações reais. Estes melhoramentos

incluem basicamente acréscimos de novas restrições e relaxação de algumas outras restrições.

Introduz-se a seguir as principais variantes do PRV.

PRV com Capacidade- PRVC

No PRVC existe uma frota de veículos idênticos, todos com certa capacidade e um único depósito. As demandas são sempre todas do depósito para um cliente (ou todas dos clientes para o depósito). O objetivo é atender as demandas respeitando a capacidade dos veículos. Cada cliente só é visitado por um único veículo. O objetivo é encontrar um conjunto de rotas com menor custo. Não há restrições adicionais sobre as rotas, essa é provavelmente a mais básica variante do PRV;

PRV com Janela de Tempo - PRVJT

O PRVJT é o mesmo problema anterior, com a restrição adicional de que cada cliente só pode ser atendido durante certo horário, chamado de janela de tempo. Essa variante exige que se conheçam os tempos de deslocamento dos veículos, bem como os tempos de atendimento em cada cliente.

PRV com Frota Heterogênea – PRVFH

No PRVFH podem existir vários tipos de veículos na frota, com diferentes capacidades, tempos de deslocamento e custos operacionais. Normalmente existem as chamadas restrições de docagem, determinados clientes não podem ser atendidos por certos tipos de veículos.

PRV com Entrega Dividida - PRVED

O PRVED é a extensão do PRVC em que é permitido que um cliente possa ser atendido por diferentes veículos se isto possibilitar uma redução do custo da rota.

PRV com Entrega e Coleta – PRVEC

No PRVEC pode haver demandas que são coletadas em um determinado cliente para serem entregues em outro cliente. As rotas ainda devem se iniciar e terminar no depósito.

PRV Multi-depósito – PRVMD

No PRVMD os veículos estão distribuídos por vários depósitos. Eventualmente uma rota pode começar em um depósito e terminar em outro.

As variantes apresentadas acima são apenas amostras dos tipos de PRV que podem ocorrer na prática. Existe um número considerável de outras restrições e combinações destas restrições que geram novas extensões. Algumas das restrições não mencionadas anteriormente: prioridade de entrega e/ou recolhimento, restrição quanto ao horário de uso de um veículo, restrições relacionadas a motoristas, taxa de ocupação mínima dos veículos, valor máximo da carga embarcada em um veículo, entre tantas outras.

2.2. Caso em Estudo

Neste trabalho definiu-se uma variante do PRV reunindo as características que são mais freqüentemente encontradas nos problemas reais. Esta variante caracteriza-se por:

- um único depósito;
- um conjunto de clientes, onde cada um possui um tipo de doca e uma demanda a ser atendida, consistindo em entrega ou coleta, e que não pode acontecer de maneira dividida;
- uma frota de veículos heterogênea, com uma quantidade máxima de veículos por tipo, capacidade, velocidade média e docagem diferenciadas, sendo os custos divididos em fixos, pelo uso de cada veículo, e variáveis, por quilômetro rodado, para cada tipo de veículo;
- os pedidos são constituídos de múltiplos produtos com diferentes quantidades, onde se trabalha com metragem cúbica para relacionar as medidas dos produtos, assim como a capacidade dos tipos de veículos;
- existem duas classes de atendimentos para tipos de clientes distintos, loja, onde pode ocorrer um entrega de um determinado produto em grande escala, ou em um cliente final, onde geralmente uma única unidade de cada determinado produto é entregue;
- os tempos de carregamento e descarregamento são fixos e devem ocorrer somente dentro de cada janela de tempo conforme cada cliente;
- todas as rotas devem ter no máximo 24 horas de percurso de tempo, onde a otimização do momento da saída do veículo deve ocorrer para que o mesmo minimize o tempo de espera na abertura de janelas;
- para cada cliente não atendido uma penalidade é atribuída à solução.

As restrições chaves que compõem o caso em estudo podem ser relacionadas como:

- Janela de Tempo;
- Frota Heterogênea;
- Docagem.

Considerando as três restrições citadas anteriormente, as mesmas destacam-se pela sua estruturação através de classes associadas a cada uma delas. Observe a seguir as tabelas que representam esta divisão, assim como a caracterização de cada classe.

Tabela 1: Descrição dos dados referentes a Janela de Tempo

JANELA DE TEMPO		
ID	TEMPO DE ABERTURA	TEMPO DE FECHAMENTO
JT10	0:00:00	23:59:59
JT20	8:00:00	12:00:00
JT30	12:00:00	20:00:00
JT40	9:00:00	20:00:00
JT50	8:00:00	19:00:00

Tabela 2: Descrição dos dados referentes a Veículos

VEICULOS					
ID	DESCRIÇÃO	QUANTIDADE	CUSTO FIXO (R\$)	CUSTO POR DISTÂNCIA (R\$/Km)	CAPACIDADE (m³)
CG	Carreta Grande	03	356	2,66	80
CM	Carreta Média	10	273	1,74	50
CP	Carreta Pequena	20	213	1,00	18

Tabela 3: Descrição dos dados referentes a Docas

DOCAS		
ID	DESCRIÇÃO	TIPO VEÍCULO
01	DockGrande	CG
02	DockMedio	CM
03	DockPeque	CP

O detalhamento dos dados referentes às três restrições indica que os dois principais objetos de um ambiente de roteamento, os locais e veículos, estão diretamente relacionados a um item de cada tabela mostrada anteriormente, que permite agrupamentos e/ou classificações dos mesmos.

A combinação dos dados referentes a cada uma das tabelas pode atribuir um grau de dificuldade ou facilidade no atendimento de um determinado cliente ou na utilização de um dado veículo, influenciando diretamente no processo de desenvolvimento da solução.

3. PACOTE ROTEAMENTO DE VEÍCULOS

A grande importância prática do PRV motivou várias companhias de *software* a desenvolverem pacotes para roteamento de veículos. As funcionalidades dos pacotes RoutePro, RouteSmart, Roadnet 500 e TruckStops, líderes de mercado nos Estados Unidos, foram analisadas por Baker [1].

Todos esses pacotes têm como filosofia oferecer o máximo de flexibilidade, no sentido em que o mesmo pacote possa ser usado em um grande número de situações. Isso é feito através de uma extensa e complexa lista de parâmetros, que permite ao usuário especificar o seu problema particular, dentre as várias possibilidades dadas. Por exemplo, o mesmo pacote pode ser usado para o planejamento de ônibus escolares ou de um serviço de entrega de encomendas.

Infelizmente, a filosofia de máxima flexibilidade impacta negativamente na qualidade das soluções obtidas. Isso porque algoritmos genéricos, a ponto de acomodar variantes do PRV tão diversas entre si, são menos eficientes do que algoritmos especializados. O cliente de um certo PRV particular certamente é penalizado por não usar os algoritmos mais adequados ao seu problema.

Além disso, como alguns clientes necessitam obter soluções de modo rápido para problemas de grande porte, com várias centenas de clientes, os pacotes optaram por só oferecer heurísticas que são garantidamente rápidas. Isso também acaba penalizando o cliente que não precisa resolver problemas tão grandes e que poderia esperar alguns minutos por uma solução de melhor qualidade.

No estudo deste trabalho utilizou-se um pacote de roteamento para análise e comparação de resultados. De modo geral estes pacotes possuem uma grande flexibilidade, permitindo a representação de diversas variantes do PRV. Ele oferece ao usuário a opção de rodar 8 diferentes heurísticas construtivas, muitas delas recebem nomes de heurísticas clássicas propostas na literatura [5] para alguma versão do PRV. Uma dessas heurísticas, Cluster, serve para ser combinada com as outras heurísticas construtivas, dando um total de 14 opções de algoritmos. Não é possível saber exatamente como essas heurísticas foram adaptadas para resolver problemas de PRV mais genéricos. Uma síntese das 8 heurísticas básicas, de acordo com informações encontradas em [2] é apresentado a seguir:

Boundary: este algoritmo constrói rotas simultaneamente. O pedido de um cliente que possui maior prioridade inicia uma rota, e assim por diante os pedidos, de acordo com sua prioridade, são selecionados. A partir do segundo pedido selecionado, o algoritmo

procura pela melhor rota, por onde este cliente possa ser atendido. Se não existir uma rota que possa atender ao pedido, uma nova rota será criada. Caso exista a possibilidade do pedido do cliente ser inserido em uma rota já iniciada, o pedido do cliente é inserido na melhor posição possível na rota. O algoritmo Boundary percorre todos os pedidos dos clientes, de acordo com sua prioridade, com o objetivo de possibilitar o atendimento do pedido através de uma rota existente ou criando uma nova rota;

Nearest Neighbor: este algoritmo constrói uma rota por vez. Uma de suas características é a utilização máxima possível de um veículo. O primeiro pedido a ser atendido em uma rota é o que possui maior prioridade. A partir do segundo pedido o critério de escolha é o da proximidade com o cliente que teve seu pedido atendido anteriormente. Quando a rota não puder atender a mais nenhum pedido, uma nova rota é inicializada;

Nearest Pass: este algoritmo possui o mesmo princípio de funcionamento do algoritmo Nearest Neighbor. O método Nearest Pass propõe a utilização máxima de um veículo antes que outro seja alocado. O primeiro cliente a ser atendido em uma rota é o que possui maior prioridade. O critério de escolha para os pedidos a serem atendidos posteriores ao primeiro será o que a somará menor custo à rota em construção. Quando a rota não puder atender a mais nenhum pedido, uma nova rota é inicializada;

Pie: este algoritmo utiliza uma estratégia similar de roteamento que a proposta pelo algoritmo Boundary. Entretanto o algoritmo Pie apresenta melhores resultados quando é aplicada na solução de um modelo que possui múltiplos depósitos, enquanto o Boundary é indicado para solucionar problemas que envolvam apenas um depósito;

Savings (Clark e Wright): este algoritmo constrói rotas simultaneamente. O algoritmo Savings (Clark e Wright) propõe inicialmente a construção de uma rota para cada pedido a ser atendido. A partir deste cenário é criada uma lista de economias que descreve o ganho a ser obtido caso haja a união de duas rotas. A lista de economias é ordenada da maior economia para a menor. Toda lista é percorrida implementando as uniões que não possuem nenhuma restrição. O processo de roteamento encerra-se após nenhuma junção de rotas ser possível;

Tree: este algoritmo constrói cada rota por vez, objetivando principalmente a máxima utilização da capacidade do veículo. O pedido com maior prioridade inicia a rota. O último cliente a ser atendido é denominado de semente. Posteriormente ao cliente semente, o

próximo cliente a ser atendido é inserido quando apresentar a menor soma de distâncias, entre a semente e o candidato, e a distância entre o depósito e o candidato, entre todos os possíveis atendimentos a serem realizados;

Weighted Constraint: este algoritmo utiliza uma estratégia similar de roteamento que a proposta pelo algoritmo Boundary e Pie. Contudo, o algoritmo Weighted Constraint é recomendado para problemas de roteamento que abordem somente capacidade de veículos, janelas de tempo, entre outras poucas;

Cluster: este algoritmo constrói uma rota por vez. Este método parte do princípio de utilizar o máximo da capacidade de um veículo. O algoritmo possui a característica de formar rotas a partir do agrupamento de um conjunto de pedidos de clientes que estejam próximos. O algoritmo Cluster apresenta melhores resultados quando ele é utilizado como um inicializador de uma solução, outros algoritmos devem ser usados para completar as rotas inicialmente oferecidas pelo Cluster.

Em termos de organização dos dados do sistema, os pacotes de roteamentos possuem uma estrutura que está definida através de objetos logísticos, como definido em [2]. A seguir uma pequena síntese dos principais objetos é disposta.

Locais

Representam todos os tipos de locais numa operação logística de desenvolvimento de rotas, que pode incluir clientes, depósitos, fábricas, centro de distribuições, fornecedores, entre outros. Dados como endereço, identificação são inerentes aos Locais.

Janelas de Tempo

Possui todas as datas e horários referentes a coleta e/ou entrega dos pedidos nos locais.

Pedidos

Representam todos os tipos de pedidos no processo de roteirização. Dados importantes que se pode citar é a quantidade produtos e o local de entrega.

Itens de Pedidos

Este objeto detalha os itens, produtos pertencentes a cada pedido, pois um único pedido pode ser composto por mais de um produto. Dados referentes a este objeto são: quantidade de produtos solicitados e de qual local os produtos são distribuídos.

Produtos

Todos os tipos de produtos disponíveis para entrega e coleta estão relacionados neste objeto. Pode-se detalhar através deste objeto o tempo de carregamento e descarregamento de cada produto.

Veículos

Descreve toda a frota de veículos e o permite criar classes (tipos) para melhor gerenciamento e representação do problema. Custo fixo, custo por distância e capacidade máxima de carga são alguns dados referentes a este objeto.

Motoristas

Este objeto relaciona todos os motoristas disponíveis para percorrer as rotas com os veículos pertencentes a frota. Pode-se associar automaticamente ou de forma manual os motoristas a cada rota.

Rotas

Este objeto tem seus dados criados automaticamente a partir da utilização dos algoritmos disponíveis no pacote. Alguns dados deste objeto são: tempo de carga e descarga em cada parada, tempo estimado de chegada, tempo estimado de partida, entre outros.

Mapas

Pode-se utilizar arquivos de mapas para melhor ambientação sobre uma perspectiva geográfica. Basicamente os pacotes disponíveis no mercado suportam dois tipos de mapas: vetoriais e digitalizados.

Existem outros objetos que podem ser utilizados, dentro da abrangência dos pacotes, como: Pedidos Encadeados; Equipes de motoristas; Perfil de Serviço; Perfis de Parada, Posicionamento de Veículos; Desempenho de Veículos; Redes de Estradas e Janelas de Hora de Rush.

O figura 2 mostra o processo básico de ações que um usuário executa no suporte logístico de criação e gerenciamento de rotas.

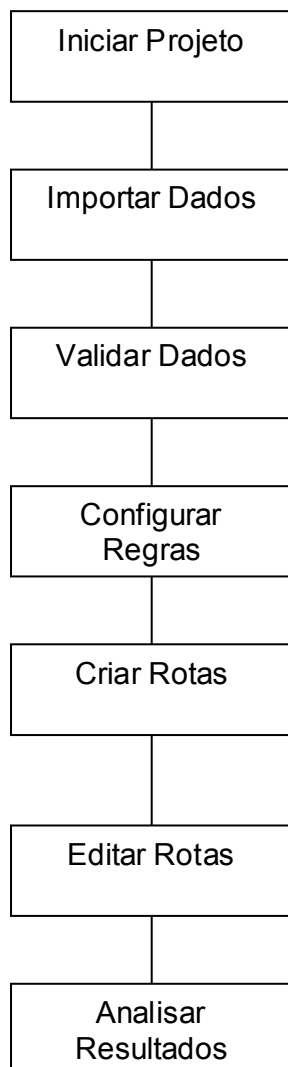


Figura 2: Processo de criação de rotas no pacote

Considerando a figura 2 pode-se descrever que o processo de roteirização tem início quando o usuário cria um novo projeto. Depois da criação de um projeto deve-se importar ou adicionar manualmente todos os dados necessários à construção de uma solução. A validação dos dados é realizada por métodos automáticos, enquanto os mesmos estão sendo inseridos ao projeto. A criação das regras consiste na parametrização do modelo que irá direcionar as ações dos algoritmos roteirizadores. A etapa de criação de rotas consiste na escolha de um método para criação de uma solução viável ao problema de roteamento. Uma vez criadas as rotas, pode-se alterá-las

com a finalidade de obter-se uma melhor solução. Os pacotes de roteamento disponibilizam para o usuário uma gama de ferramentas que possibilitam a análise da solução obtida.

4. Heurísticas Construtivas

O fato de um problema ser classificado como NP-difícil, caso em que o Problema de Roteamento de Veículos se encontra, é aceito como forte justificativa para o uso de algoritmos que optem por abdicar da certeza da obtenção da melhor solução em troca de um tempo de execução polinomial. Naturalmente se deseja encontrar soluções de boa qualidade, próximas da ótima. Esses algoritmos são conhecidos como heurísticas. Pode-se usar esse tipo de algoritmo também em conjunto com outras técnicas exatas, como por exemplo, a programação inteira.

Uma importante subclasse de heurísticas são as chamadas heurísticas construtivas, aquelas que tentam construir uma solução para um problema a partir apenas de sua definição. Muitas dessas heurísticas constroem essas soluções tomando uma seqüência de decisões encadeadas, em cada passo tomando a decisão que menos aumenta o custo da solução. Essas heurísticas são chamadas de gulosas ou míopes. Como a máxima economia de custo em um determinado passo pode levar a custos maiores nos passos seguintes, tais heurísticas muitas vezes levam as soluções de baixa qualidade.

Implementação de duas das heurísticas construtivas mais clássicas para o PRV: Vizinho mais Próximo e Economias. Os algoritmos foram adaptados para o modelo de PRV proposto.

4.1. Vizinho mais Próximo

A heurística míope do Vizinho mais Próximo, na língua inglesa denominada *Nearest Neighbor*, propõe a construção de cada rota uma por vez, ou seja, de forma serial. A versão proposta por este trabalho foi construída e adaptada para a variante do PRV estudada, e consiste nas ilustrações e descrições a seguir:

Passo 1: inicialmente se tem um conjunto de demandas (em laranja) para cada cliente (círculo), assim como um custo de viagem entre os clientes (em vermelho), inclusive o depósito (quadrado), e um conjunto de veículos heterogêneo (custos e capacidade de carga) que tem como origem o próprio depósito, devendo retornar para o depósito ao final da rota;

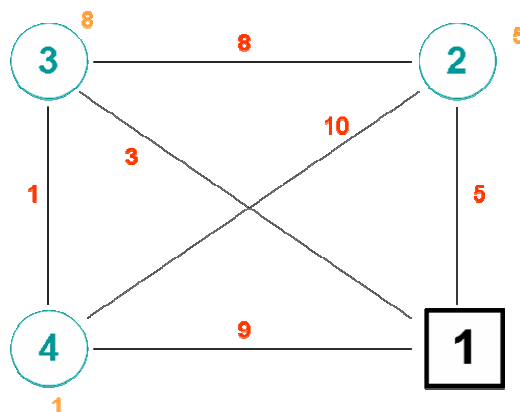


Figura 3: Representação dos clientes (demandas), depósito e custos de distância

Passo 2: a cada iteração iniciar uma nova rota a partir do depósito, com o veículo de menor custo fixo pertencente a frota de veículos, que ainda não foi utilizado e se objetiva encontrar o cliente, referente ao atendimento, mais próximo, referente à distância com o cliente atendido anteriormente ou depósito, que não viole as restrições de docagem, janela de tempo e do veículo, e possui uma demanda de pedido não atendida ;

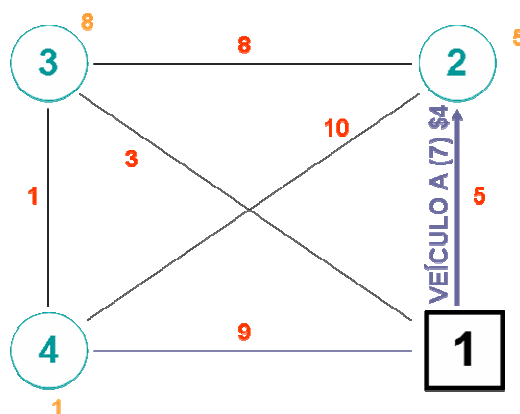


Figura 4: Veículo inicia rota saindo do depósito para o cliente viável mais próximo

Passo 3: após iniciar a rota e atender o primeiro cliente, procura-se o que possui demanda e ainda não foi atendido, que seja o mais próximo do local do último atendimento, e que não tenha nenhuma restrição empregada pelo modelo de PRV proposto;

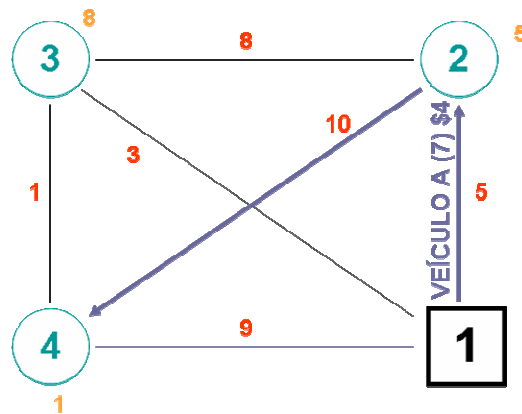


Figura 5: Visita ao segundo cliente viável mais próximo, a partir da primeira visita

Passo 4: a rota é encerrada quando não existem mais atendimentos a serem realizados ou as restrições do problema (caso do exemplo ilustrativo) impeçam que mais algum atendimento seja realizado pelo veículo da rota atual, neste caso se encerra com o veículo retornando para o depósito diretamente do último cliente visitado;

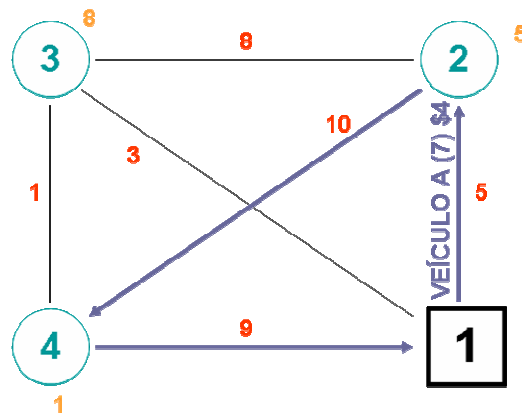


Figura 6: Não existindo mais atendimentos viáveis, o veículo retorna ao depósito

Passo 5: todo o processo é repetido para a criação de uma nova rota, até que todas as demandas tenham sido atendidas ou não mais existirem veículos disponíveis na frota, então se encerra a execução da heurística.

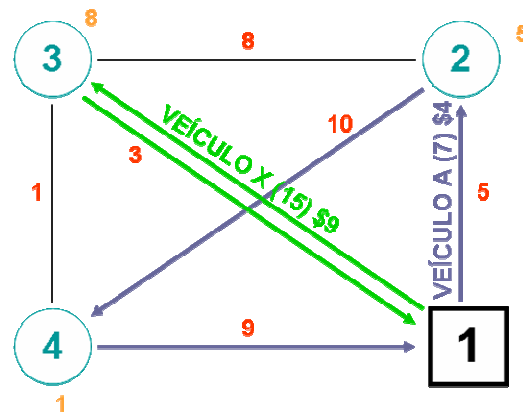


Figura 7: Solução final após a execução da lógica do Vizinho mais Próximo

O pseudocódigo da heurística do Vizinho mais Próximo implementada por este trabalho é o seguinte:

01.	HEURÍSTICA vizinho mais próximo;
02.	
03.	ENTRADA: instância;
04.	SAÍDA : instância atualizada + solução viável;
05.	
06.	INÍCIO
07.	INICIAR solução;
08.	ORDERNAR veículos pelo menor custo fixo;
09.	ENQUANTO número de pedidos < total de pedidos E existirem veículos disponíveis FAÇA
10.	INICIAR rota;
11.	atendimento O = depósito;
12.	ENQUANTO existir atendimentos possíveis de serem visitados FAÇA
13.	menor = $+\infty$;
14.	PARA todos os atendimentos K não roteados E atendimento K \neq depósito FAÇA
15.	SE distância(atendimento O, atendimento K) < menor ENTÃO
16.	SE doca(s) do veículo é compatível com doca do atendimento K ENTÃO
17.	SE tempo viagem(atendimento O, atendimento K)
18.	< fechamento janela tempo do atendimento K ENTÃO
19.	SE restrições do veículo são respeitadas ENTÃO
20.	menor = distância(atendimento O, atendimento K);
21.	FIM-SE;
22.	FIM-SE;
23.	FIM-SE;
24.	FIM-SE;
25.	FIM-PARA;
26.	SE menor < $+\infty$ ENTÃO
27.	ADICIONAR atendimento K a rota;
28.	atendimento O = atendimento K;
29.	FIM-SE;
30.	FIM-ENQUANTO;
31.	SE rota possui atendimentos ENTÃO
32.	ADICIONAR rota;
33.	FIM-SE;
34.	FIM-ENQUANTO;
35.	FIM.

Figura 8: Pseudocódigo Heurística Vizinho mais Próximo

4.2. Economias

A heurística de Economias foi sugerida originalmente em [19], também conhecida como *Savings* (no idioma inglês), e inicialmente foi proposta para o Problema de Roteamento de Veículos com Capacidade. Baseia-se na noção de economias, que pode ser definido como o custo da combinação, ou união, de duas rotas existentes. Trata-se de uma heurística iterativa de construção baseada numa função gulosa de inserção.

A heurística de Economias possui duas variantes distintas, a paralela e a serial. A variante serial consiste na construção de uma rota por vez, enquanto a paralela cria várias rotas ao mesmo tempo. A proposta deste trabalho é utilizar como base a versão paralela.

Uma variante genérica da heurística de Economias paralela pode ser descrita através dos seguintes descrições e ilustrações a seguir:

Passo 1: inicialmente se tem um conjunto de demandas (em laranja) para cada cliente (círculo), assim como um custo de viagem entre os clientes (em vermelho), inclusive o depósito (quadrado), e um conjunto de veículos heterogêneo (custos e capacidade de carga) que tem como origem o próprio depósito, devendo retornar para o depósito ao final da rota;

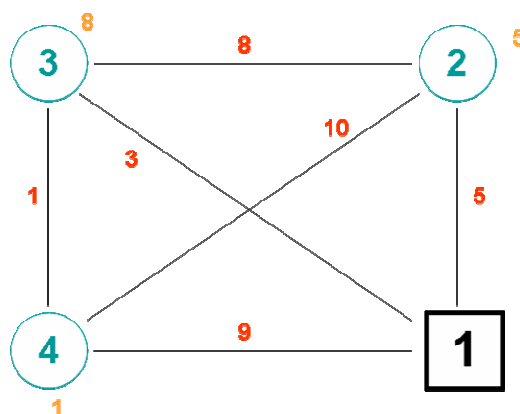


Figura 9: Representação dos clientes (demandas), depósito e custos de distância

Passo 2: cria-se uma solução onde todos as demandas (clientes) são atendidos por um veículo, constituindo assim rotas de ida e volta, ou seja, ciclos de tamanho 2, juntamente com uma lista de economias, que consiste em todas as possíveis combinações viáveis das rotas existentes, ordenado-as de forma decrescente de acordo com o custo de cada

combinação (maior economia para a menor economia), onde C_{ij} é o custo da união das rotas indo do cliente i para o j , estando os clientes i e j em rotas diferentes;

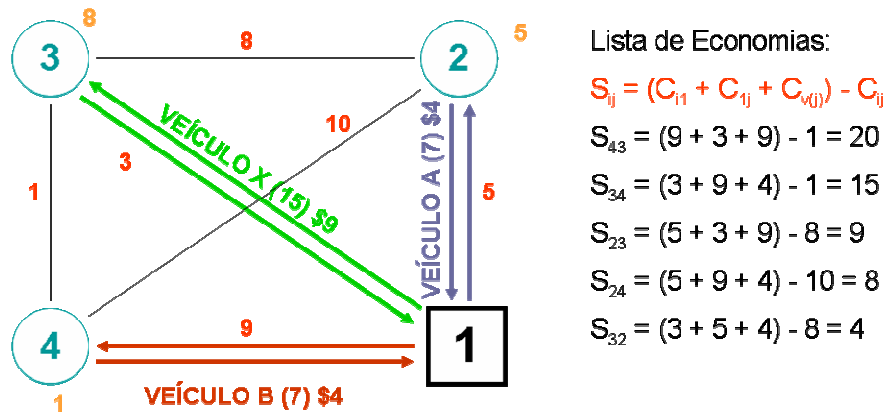


Figura 10: Conjunto de rotas inicialmente criadas com a lista de economias gerada

Passo 3: percorrer toda a lista de economias procurando minimizar o custo da solução a partir de cada combinação viável, o ganho na economia será a diferença entre o percurso anterior e o novo. Atualizar a lista de economias com a finalidade de encontrar alguma nova combinação que possa gerar economias;

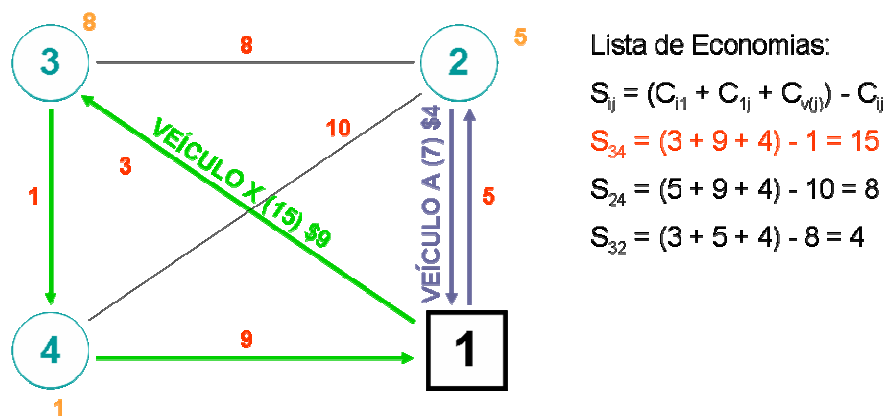


Figura 11: União das rotas, a partir da economia gerada com os clientes 3 e 4

Passo 5: quando não forem mais encontradas economias a partir das uniões de duas rotas finalizar a heurística ou não existirem mais uniões viáveis, encerra-se a heurística de Economias.

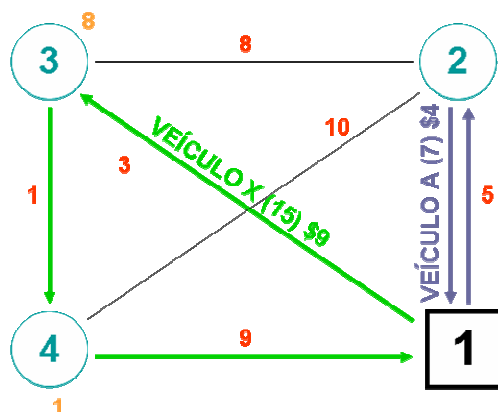


Figura 12: Solução final após a execução da lógica da heurística de Economias

A partir desta idéia inicial apresentada anteriormente este trabalho propõe duas versões da heurística de Economias para o modelo proposto de Problema de Roteamento de Veículos. A base da diferença entre estas duas versões está no momento e quantidade de atualizações que a lista de economias sofrerá (passo 4), impactando diretamente no tempo de execução e na qualidade da solução.

Uma importante adição às duas versões propostas da idéia inicial apresentada anteriormente é a alocação de veículos quando as rotas não estão mais sujeitas à modificação, pois durante todo o processo do algoritmo de economias os veículos são repetidos para mais de uma rota.

4.2.1. Versão 1

A primeira versão da heurística de Economias consiste em percorrer toda a lista de economias buscando por combinações viáveis. Assim que o algoritmo verificar todas as opções da lista, a mesma é atualizada, e assim sucessivamente até que uma lista de economia não possua mais uniões viáveis que gerem economias.

O pseudocódigo desta versão da heurística de Economias, implementada por este trabalho, é o seguinte:

01.	HEURÍSTICA economias – versão 1;
02.	
03.	ENTRADA: instância;
04.	SAÍDA : instância atualizada + solução viável;
05.	
06.	INÍCIO
07.	CRIAR SOLUÇÃO INICIAL;
08.	CRIAR LISTA DE ECONOMIAS;
09.	ENQUANTO existirem economias viáveis FAÇA
10.	PARA todas as economias da lista de economias, iniciando pela de maior ganho FAÇA
11.	SE união das rotas for viável ENTÃO
12.	UNIR ROTAS;
13.	FIM-SE;
14.	FIM-PARA;
15.	ATUALIZAR LISTA DE ECONOMIAS;
16.	FIM-ENQUANTO;
17.	ALOCAR VEICULOS;
18.	CALCULAR custo solução;
19.	FIM.

Figura 13: Pseudocódigo Heurística de Economias – versão 1

4.2.2. Versão 2

A versão 2 se propõe percorrer toda a lista de economias até seu final ou até que uma união de rotas na lista (economia) não possa ser realizada. A cada nova união realizada a lista de economias é atualizada. Isto torna o algoritmo mais lento em relação à primeira versão, pois um maior número de atualizações da lista de economias ocorre, mas possibilita uma maior exploração em relação às possibilidades de combinações de rotas que possam proporcionar maior ganho.

O pseudocódigo desta versão da heurística de Economias, implementada por este trabalho, é o seguinte:

01.	HEURÍSTICA economias – versão 2;
02.	
03.	ENTRADA: instância;
04.	SAÍDA : instância atualizada + solução viável;
05.	
06.	INÍCIO
07.	CRIAR SOLUÇÃO INICIAL;
08.	CRIAR LISTA DE ECONOMIAS;
09.	ENQUANTO existirem economias viáveis FAÇA
10.	PARA todas as economias da lista de economias, iniciando pela de maior ganho
11.	E que não seja encontrada uma economia que não possa ser aplicada FAÇA
12.	UNIR ROTAS;
13.	FIM-PARA;
14.	ATUALIZAR LISTA DE ECONOMIAS;
15.	FIM-ENQUANTO;
16.	ALOCAR VEICULOS;
17.	CALCULAR custo solução;
18.	FIM.

Figura 14: Pseudocódigo Heurística Economias – versão 2

5. Buscas Locais

Algoritmos de busca local são heurísticas para aprimorar soluções encontradas por outros métodos, freqüentemente heurísticas construtivas. Estes aperfeiçoamentos são obtidos através da idéia de busca em vizinhanças. Uma vizinhança é um subconjunto das soluções viáveis que podem ser obtidas a partir da aplicação de uma operação pré-definida sobre uma solução base dada.

As questões fundamentais para construir-se uma busca local são:

- definição da vizinhança;
- estratégias de busca na vizinhança;
- complexidade de cada iteração.

Nas heurísticas de busca local propostas por este trabalho é necessário possuir dados referente aos limites de tempo (horário) em que ocorrem ou poderá ocorrer os atendimentos para cada cliente, indicando: tempo estimado de chegada aos clientes atendidos (ETA – Estimated Time Arrival); tempo estimado de saída dos clientes atendidos (ETD – Estimated Time of Departure); último tempo de partida de cada cliente atendido (LTD – Late Terminal Departure); e o último tempo de chegada a um cliente (LTA – Late Terminal Arrival). Estes dados relacionados ao locais de cada atendimento têm como objetivo minimizar o tempo de inserção de um novo atendimento, em uma rota já constituída, em qualquer posição na ordem existente de atendimentos. Os valores de tempo são definidos a partir de cálculos que envolvem tempo de saída do veículo do depósito, tempo de viagem, tempo de espera de abertura de janelas, caso ocorra, tempo de descarregamento da mercadoria referente ao pedido, e tempos referentes a abertura e fechamento do horário para atendimento à cada cliente.

A seguir são descritas as buscas locais propostas por este trabalho.

5.1. Eliminar Rotas

Esta busca local tem como objetivo a redução do número de rotas. A eliminação de uma rota acontece quando todas as demandas atendidas por uma determinada rota passam a ser atendidas por outras rotas já existentes. Diminuindo o número de veículos na solução do problema, eliminam-se os custos fixos correspondentes. A busca local Eliminar Rotas consiste nos seguintes passos:

Passo 1: a cada iteração objetiva-se a eliminação de uma rota, iniciando-se pela rota com menor número de atendimentos até a metade da quantidade total de rotas, sendo a lista de rotas ordenada pelo número de pedidos atendidos de cada rota;

Passo 2: para todos os atendimentos da rota que se deseja eliminar, procurar um local em uma segunda rota em que se possa inserir este atendimento, respeitando as restrições impostas pelo modelo do problema;

Passo 3: a escolha da rota onde se tenta a inserção do atendimento é dada através da disposição das rotas candidatas em uma lista, onde a mesma esta organizada em ordem decrescente do número de atendimentos de cada rota;

Passo 4: se todos os atendimentos da rota possam ser inseridos em outras rotas elimina-se a rota, caso contrário desconsidera-se a mudança dos atendimentos;

Passo 5: quando metade do total de rotas da solução for verificada em relação a sua eliminação, encerrar a busca local.

O pseudocódigo do Eliminar Rotas implementada neste trabalho é o seguinte:

01.	BUSCA LOCAL eliminar rotas;
02.	
03.	ENTRADA: solução viável + instância;
04.	SAÍDA : solução viável atualizada + instância atualizada;
05.	
06.	INÍCIO
07.	ORDERNAR lista de rotas por ordem crescente, quantidade de atendimentos por rota;
08.	CRIAR BACKUP da lista de rotas;
09.	PARA metade das rotas P, iniciando pela rota com menor número de atendimentos FAÇA
10.	PARA todos os atendimentos K pertencentes a rota P ≠ depósito FAÇA
11.	PARA todas as rotas P' ≠ P, iniciando pela rota com maior número de atendimentos
12.	E atendimento K não foi retirado da rota P FAÇA
13.	SE doca(s) do veículo da rota P' é compatível com doca do atendimento K ENTÃO
14.	PARA todos os atendimentos L pertencentes a rota P' ≠ depósito
15.	E atendimento K não foi retirado da rota P FAÇA
16.	SE tempo viagem(atendimento L-1, atendimento K)
17.	< fechamento janela tempo do atendimento K ENTÃO
18.	SE restrições do veículo são respeitadas ENTÃO
19.	SE tempo viagem(atendimento L-1, atendimento K, atendimento L)
20.	≤ valor do LTA do atendimento L ENTÃO
21.	ADICIONAR atendimento K na rota P' entre atendimento L-1 e L;
22.	REMOVER atendimento K da rota P;
23.	CALCULAR tempo de saída de cada atendimento da rota P;
24.	CALCULAR tempo de saída de cada atendimento da rota P';
25.	CALCULAR custo da rota P;
26.	CALCULAR custo da rota P';
27.	FIM-SE;
28.	FIM-SE;
29.	FIM-SE;
30.	FIM-PARA;
31.	FIM-SE;
32.	FIM-PARA;
33.	FIM-PARA;
34.	SE todos os atendimentos k foram removidos de P ENTÃO
35.	REMOVER rota P;
36.	CALCULAR custo solução;
37.	CRIAR BACKUP da lista de rotas;
38.	SENÃO
39.	RESTAURAR backup da lista de rotas;
40.	FIM-SE;
41.	FIM-PARA;
42.	FIM.

Figura 15: Pseudocódigo Busca Local Eliminar Rotas

5.2. Mudar Atendimentos

Esta busca local experimenta mudança no atendimento de uma determinada rota para uma segunda rota, na tentativa de obter uma diminuição no custo total da solução. Caso uma rota tenha todos seus atendimentos transferidos para outras rotas, esta rota será eliminada. Um atendimento pode ser movido de uma rota para outra somente uma vez. Os passos desta busca local são descritos a ser seguir:

Passo 1: a cada iteração objetiva-se a retirada de um atendimento de uma rota para uma outra, a escolha da rota, em que começará o diagnóstico para a mudança dos atendimentos, será da rota com menor número de atendimentos para a de maior número de atendimentos;

Passo 2: para todos os atendimentos, em suas respectivas rotas, que se deseja retirar da rota original e inserir em uma segunda, procurar um local em que se possa adicionar este atendimento, respeitando as restrições impostas pelo modelo do problema, que seja gerada uma redução no custo total da solução e que o atendimento de pedido não tenha sido anteriormente já modificado de uma rota em uma outra iteração, e incluí-la no novo local;

Passo 3: a escolha da rota onde se tenta a inserção do atendimento é dada através da disposição das rotas candidatas em uma lista, onde a mesma está organizada em ordem decrescente do número de atendimentos de cada rota;

Passo 4: caso todos os atendimentos da rota possam ser inseridos em outras rotas elimina-se a rota, gerando mais uma forma de economia para a solução;

Passo 5: quando todos os atendimentos das rotas da solução forem verificados em relação à troca de rotas, encerra-se a busca local.

O pseudocódigo do Mudar Atendimentos implementado neste trabalho é o seguinte:

01.	BUSCA LOCAL mudar atendimentos;
02.	
03.	ENTRADA: solução viável + instância;
04.	SAÍDA : solução viável atualizada + instância atualizada;
05.	
06.	INÍCIO
07.	
08.	ORDERNAR lista de rotas por ordem decrescente, quantidade de atendimentos por rota;
09.	PARA todas as rotas P, iniciando pela rota com maior número de atendimentos FAÇA
10.	PARA todos os atendimentos K pertencentes a rota P ≠ depósito FAÇA
11.	PARA todas as rotas P' ≠ P, iniciando pela rota com maior número de atendimentos
12.	E atendimento K não foi trocado de rota FAÇA
13.	SE doca(s) do veículo da rota P' é compatível com doca do atendimento K ENTÃO
14.	PARA todos os atendimentos L pertencentes a rota P' ≠ depósito FAÇA
15.	SE tempo viagem(atendimento L-1, atendimento K)
16.	< fechamento janela tempo do atendimento K ENTÃO
17.	SE restrições do veículo da rota P' são respeitadas ENTÃO
18.	SE tempo viagem(atendimento L-1, atendimento K, atendimento L)
19.	≤ valor do LTA do atendimento L ENTÃO
20.	ADICIONAR atendimento K na rota P' entre atendimento L-1 e L;
21.	REMOVER atendimento K da rota P;
22.	CALCULAR tempo de saída de cada atendimento da rota P;
23.	CALCULAR tempo de saída de cada atendimento da rota P';
24.	CALCULAR custo da rota P;
25.	CALCULAR custo da rota P';
26.	CALCULAR novo custo da solução;
27.	SE novo custo da solução < antigo custo da solução ENTÃO
28.	CALCULAR valor do LTA de cada atendimento da rota P;
29.	CALCULAR valor do LTA de cada atendimento da rota P';
30.	ABANDONAR repetição;
31.	SENÃO
32.	DESFAZER alterações na rota P e P';
33.	CALCULAR tempo de saída de cada atendimento da rota P;
34.	CALCULAR tempo de saída de cada atendimento da rota P';
35.	CALCULAR custo da rota P;
36.	CALCULAR custo da rota P';
37.	CALCULAR custo da solução;
38.	FIM-SE;
39.	FIM-SE;
40.	FIM-SE;
41.	FIM-SE;
42.	FIM-PARA;
43.	FIM-SE;
44.	FIM-PARA;
45.	FIM-PARA;
46.	SE todos os atendimentos K foram removidos de P ENTÃO
47.	REMOVER rota P;
48.	FIM-SE;
49.	FIM-PARA;
50.	FIM.

Figura 16: Pseudocódigo Busca Local Mudar Atendimentos

5.3. Trocar Atendimentos

Esta busca local experimenta a troca de dois atendimentos em duas rotas diferentes. Um atendimento só pode ser trocado de rota uma única vez.

Os passos desta busca local são descritos a seguir:

Passo 1: a cada iteração objetiva-se a troca de um atendimento de uma rota por outro atendimento posicionado em uma segunda rota.

Passo 2: escolhe-se o primeiro atendimento que fará parte da troca, a partir da posição do mesmo na rota, em ordem seqüencial que ocorrem as visitas. As rotas investigadas para a escolha do primeiro atendimento são exploradas em ordem decrescente, de acordo com o maior número de atendimentos para a rota com menor número de atendimentos realizados;

Passo 3: escolhe-se o segundo atendimento que fará parte da troca, a partir da posição do mesmo na rota, em ordem seqüencial que ocorrem as visitas. As rotas investigadas para a escolha do segundo atendimento são exploradas em ordem decrescente, de acordo com o maior número de atendimentos para a rota com menor número de atendimentos realizados, diferente da rota do primeiro atendimento escolhido (passo 2);

Passo 4: para todos os atendimentos, em suas respectivas rotas, que se deseja trocar por um outro atendimento de uma rota diferente, procurar um local em que esteja o segundo atendimento e se possa trocar pelo primeiro atendimento, respeitando as restrições impostas pelo modelo do problema, que seja gerada uma redução no custo total da solução e que nenhum dos atendimentos de pedido envolvidos na troca já tenham sido trocados em uma outra iteração, e então realize a troca;

Passo 5: quando todos os atendimentos das rotas da solução forem verificados em relação à troca de atendimentos, encerrar a busca local.

A seguir o pseudocódigo da busca local Trocar Atendimentos é apresentado:

01.	BUSCA LOCAL trocar atendimentos;
02.	
03.	ENTRADA: solução viável + instância;
04.	SAÍDA : solução viável atualizada;
05.	
06.	INÍCIO
07.	
08.	ORDERNAR lista de rotas por ordem decrescente, quantidade de atendimentos por rota;
09.	PARA todas as rotas P, iniciando pela rota com maior número de atendimentos FAÇA
10.	PARA todos os atendimentos K pertencentes a rota P ≠ depósito FAÇA
11.	PARA todas as rotas P' ≠ P, iniciando pela rota com maior número de atendimentos
12.	E atendimento K não foi trocado de rota FAÇA
13.	SE doca(s) do veículo da rota P é compatível com doca(s) do veiculo da rota P'
14.	ENTÃO
15.	PARA todos os atendimentos L pertencentes a rota P' ≠ depósito FAÇA
16.	SE tempo viagem(atendimento L-1, atendimento K)
17.	< fechamento janela tempo do atendimento L+1 E
18.	tempo viagem(atendimento K-1, atendimento L)
19.	< fechamento janela tempo do atendimento K+1 ENTÃO
20.	SE restrições do veículo da rota P são respeitadas E
21.	restrições do veículo da rota P' são respeitadas
22.	ENTÃO
23.	SE tempo viagem(atendimento L-1, atendimento K, atendimento L+1)
24.	≤ valor LTA do atendimento L+1 E
25.	tempo viagem(atendimento K-1, atendimento L, atendimento K+1)
26.	≤ valor LTA do atendimento K+1
27.	ENTÃO
28.	TROCAR atendimento K na rota P por atendimento L na rota P';
29.	CALCULAR tempo de saída de cada atendimento da rota P;
30.	CALCULAR tempo de saída de cada atendimento da rota P';
31.	CALCULAR custo da rota P;
32.	CALCULAR custo da rota P';
33.	CALCULAR novo custo da solução;
34.	SE novo custo da solução < antigo custo da solução ENTÃO
35.	CALCULAR valor do LTA de cada atendimento da rota P;
36.	CALCULAR valor do LTA de cada atendimento da rota P';
37.	ABANDONAR repetição;
38.	SENÃO
39.	DESFAZER alterações na rota P e P';
40.	CALCULAR tempo de saída de cada atendimento da rota P;
41.	CALCULAR tempo de saída de cada atendimento da rota P';
42.	CALCULAR custo da rota P;
43.	CALCULAR custo da rota P';
44.	CALCULAR custo da solução;
45.	FIM-SE;
46.	FIM-SE;
47.	FIM-SE;
48.	FIM-SE;
49.	FIM-PARA;
50.	FIM-SE;
51.	FIM-PARA;
52.	FIM-PARA;
53.	FIM-PARA;
54.	FIM.

Figura 17: Pseudocódigo Busca Local Trocar Atendimentos

5.4. Resequenciar Atendimentos

Esta busca local tenta trocar a ordem dos atendimentos em uma rota na tentativa de reduzir o custo. Um atendimento só pode ser trocado, na ordem de atendimento, uma única vez. Uma breve descrição segue:

Passo 1: a cada iteração objetiva-se a realocação de um atendimento em um novo local em sua própria rota;

Passo 2: para todos os atendimentos aos pedidos, iniciando pelo primeiro atendimento realizado até o último, em todas as rotas, procurar um novo local dentro da própria rota que respeite as restrições impostas pelo problema, que proporcione uma redução no custo e não tenha ainda sido realocado;

Passo 3: quando todos os atendimentos das rotas da solução forem verificados em relação à operação de realocação, encerrar a busca local.

Em seguida o pseudocódigo a busca local é mostrada:

01.	BUSCA LOCAL resequenciar atendimentos;
02.	
03.	ENTRADA: solução viável + instância;
04.	SAÍDA : solução viável atualizada;
05.	
06.	INÍCIO
07.	
08.	PARA todas as rotas P FAÇA
09.	PARA todos os atendimentos K pertencentes a rota P ≠ depósito
10.	E atendimento K não modificado na seqüência FAÇA
11.	custo original = custo da rota P;
12.	REMOVER atendimento K;
13.	CALCULAR tempo de saída de cada atendimento da rota P;
14.	CALCULAR valor do LTA de cada atendimento da rota P;
15.	PARA todos os atendimentos L pertencentes a rota P ≠ depósito
16.	E atendimento K ≠ atendimento original FAÇA
17.	ADICIONAR atendimento K após atendimento L;
18.	SE tempo viagem (atendimento L, atendimento K)
19.	< valor do LTA do atendimento L+1 ENTÃO
20.	CALCULAR tempo de saída de cada atendimento da rota P;
21.	CALCULAR custo da rota P;
22.	SE custo atual < custo original ENTÃO
23.	melhor posição = atendimento L;
24.	FIM-SE;
25.	FIM-SE;
26.	REMOVER atendimento K da rota P;
27.	CALCULAR tempo de saída de cada atendimento da rota P;
28.	CALCULAR valor do LTA de cada atendimento da rota P;
29.	CALCULAR custo da rota P;
30.	FIM-PARA;
31.	SE houve alguma melhoria na troca de lugar do atendimento K ENTÃO
32.	ADICIONAR atendimento K após melhor posição;
33.	SENÃO
34.	ADICIONAR atendimento K no lugar original;
35.	FIM-SE;
36.	CALCULAR tempo de saída de cada atendimento da rota P;
37.	CALCULAR valor do LTA de cada atendimento da rota P;
38.	CALCULAR custo da rota P;
39.	CALCULAR custo da solução;
40.	FIM-PARA;
41.	FIM-ENQUANTO;
42.	FIM.

Figura 18: Pseudocódigo Busca Local Resequenciar Atendimentos

5.5. Inserir Atendimentos

Devido a limitações nas quantidades de veículos disponíveis, é possível que os algoritmos construtivos não consigam rotear todos os clientes, o que implica numa penalização. Esta busca local tenta inserir clientes não-atendidos para alguma rota já existente, com um procedimento análogo ao Eliminar Rotas. Observe um resumo de seus passos:

Passo 1: a cada iteração objetiva-se a locação de um atendimento não realizado em um local de uma determinada rota;

Passo 2: para todos os atendimentos não efetuados, buscar um local em uma rota, iniciando pela rota com menor número de atendimentos para a de maior número, que se possa inserir o atendimento sem infringir as restrições impostas pelo problema;

Passo 3: quando todos os atendimentos não realizados forem verificados em relação à operação de locação, encerrar a busca local.

O pseudocódigo do Inserir Atendimentos segue:

01.	BUSCA LOCAL inserir atendimentos;
02.	
03.	ENTRADA: solução viável + instância + lista de atendimentos não roteados;
04.	SAÍDA : solução viável atualizada + instância atualizada;
05.	
06.	INÍCIO
07.	ORDERNAR lista de rotas por ordem crescente, quantidade de atendimentos por rota;
08.	PARA todos os atendimentos K não roteados FAÇA
09.	PARA todas as rotas P e atendimento K não foi inserido em uma rota P FAÇA
10.	SE doca(s) do veículo da rota P é compatível com doca do atendimento K ENTÃO
11.	PARA todos os atendimentos L pertencentes a rota P FAÇA
12.	SE tempo viagem(atendimento L, atendimento K)
13.	< fechamento janela tempo do atendimento K ENTÃO
14.	SE restrições do veículo da rota P são respeitadas ENTÃO
15.	SE tempo viagem(atendimento L, atendimento K, atendimento L+1)
16.	≤ valor do LTA do atendimento L ENTÃO
17.	ADICIONAR atendimento K na rota P entre atendimento L e L+1;
18.	CALCULAR tempo de saída de cada atendimento da rota P;
19.	CALCULAR novo custo da solução;
20.	FIM-SE;
21.	FIM-SE;
22.	FIM-SE;
23.	FIM-PARA;
24.	FIM-SE;
25.	FIM-PARA;
26.	FIM-PARA;
27.	FIM.

Figura 19: Pseudocódigo Busca Local Inserir Atendimentos

6. Metaheurística GRASP

Pode-se definir uma metaheurística como um processo iterativo que direciona uma heurística subordinada em uma metodologia de busca por vizinhanças se utilizando de métodos de aprendizagem para escapar de ótimos locais e encontrar, ou chegar o mais próximo possível, do ótimo global, [22], [23], [24], [25] e [26]. Estas metodologias de investigação devem procurar um balanceamento entre uma busca intensificada e a exploração de espaço diversificados. Este equilíbrio é necessário para se identificar rapidamente regiões dentro do espaço de busca com soluções de alta qualidade, evitando perder muito tempo em regiões de busca já exploradas e que demonstrem não serem promissoras.

Algumas das principais metaheurísticas são: Busca Tabu, Algoritmos Genéticos, VNS, GRASP, *Simulated Annealing*, entre outras.

A metaheurística GRASP (*Greedy Random Adaptive Search Procedure*) foi proposta em [04]. O GRASP é uma metaheurística *multi-start*, significando que entre uma iteração e outra, todo o processo de construção e refinamento de uma solução é reiniciado, que torna as iterações independentes. O número de iterações é um parâmetro que deve ser configurado de acordo com a estratégia de abordagem do problema em estudo. O GRASP basicamente consiste em duas fases: fase de construção e busca local.

Na primeira fase, chamada de fase de construção, uma solução viável é criada elemento por elemento. Em cada iteração é feita uma escolha numa lista de candidatos restritos (LCR) por uma função gulosa que mede a qualidade cada elemento. A maneira de se selecionar os candidatos é aleatória, sendo que o grau de aleatoriedade é determinado através de um parâmetro conhecido como α . O valor de α fica entre 0 e 1, sendo que quanto mais se aproxima do valor 0, a escolha do elemento na LCR é mais gulosa, sendo que se o valor de α se aproximar de 1 a escolha é mais aleatória. Sendo se α for igual a 0 a escolha é totalmente gulosa, tendendo para que a melhor opção da LCR seja escolhida, de acordo com a função objetiva de otimização, caso α seja 1, a escolha é totalmente aleatória. A seleção gulosa implica em soluções finais de baixa diversidade, enquanto em uma escolha com nível de aleatoriedade alto a qualidade das soluções são inferiores e assim tornando a segunda fase (busca local) mais lenta.

A segunda fase é denominada de fase de busca local que consiste em explorar a vizinhança da solução obtida na fase anterior. A solução atual é atualizada e uma nova

busca na vizinhança é iniciada com uma nova solução. Basicamente o processo se repete até que nenhum melhoramento seja obtido.

Este trabalho apresenta duas versões de heurísticas baseadas na metaheurística GRASP para a variante do Problema de Roteamento de Veículos proposta. Pode-se encontrar trabalhos na literatura utilizando o GRASP em outras variantes do PRV, como em [13] e [18]. As duas versões propostas se utilizam das buscas locais já descritas no Capítulo 5 e na fase de construção é adaptado a heurística Vizinho mais Próximo para a primeira proposta de GRASP e a heurística de Economias – versão 2, a segunda proposta.

A estrutura principal do pseudocódigo para as duas versões é idêntica, a não ser pelo comando que executa a fase de construção, sendo assim segue o código:

01.	METAHEURISTICA GRASP;
02.	
03.	ENTRADA: instância + número de iterações + alpha;
04.	SAÍDA : solução viável atualizada + instância atualizada;
05.	
06.	INÍCIO
07.	CONSTRUTIVO (Vizinho mais Próximo/Economias – versão 2) + alpha;
08.	ELIMINAR ROTAS;
09.	RETIRAR ATENDIMENTOS;
10.	TROCAR ATENDIMENTOS;
11.	RESEQUENCIAR ATENDIMENTOS;
12.	SE houve atendimentos não realizados ENTÃO
13.	INSERIR ATENDIMENTOS;
14.	FIM-SE;
15.	PARA 1 até o número de iterações - 1 FAÇA
16.	CONSTRUTIVO (Vizinho mais Próximo/Economias versão 2) + alpha;
17.	ELIMINAR ROTAS;
18.	RETIRAR ATENDIMENTOS;
19.	TROCAR ATENDIMENTOS;
20.	RESEQUENCIAR ATENDIMENTOS;
21.	SE houve atendimentos não realizados ENTÃO
22.	INSERIR ATENDIMENTOS;
23.	FIM-SE;
24.	SE nova solução < melhor solução ENTÃO
25.	ATUALIZA SOLUÇÃO;
26.	FIM-SE;
27.	FIM-PARA;
28.	FIM.

Figura 20: GRASP para o Problema Roteamento de Veículos

Na primeira fase foi utilizada uma extensão da heurística construtiva Vizinho mais Próximo/Economias versão 2. Independente das diferentes estratégias apresentadas em cada uma das heurísticas construtivas, ambas possuem algumas características similares para poderem ser utilizadas na fase de construção da metaheurística GRASP. A

diferença destas novas versões para o GRASP e as originais está na escolha do próximo cliente a ser atendido em cada rota. Ao invés de se escolher o cliente que agregue menor valor a solução, ou seja, o mais barato, cria-se uma lista de candidatos (LCR) composta por todos os clientes possíveis de serem adicionados. A escolha do próximo cliente será aleatória, mas apenas entre um subgrupo dos candidatos viáveis, que possuam menor custo a ser somado a solução. A quantidade de candidatos será determinada de acordo com o parâmetro alpha, que determinar um corte (filtro) entre os possíveis candidatos. Um candidato é filtrado através de seu custo, de acordo com uma comparação com o valor de corte, que é obtido através da seguinte fórmula: candidato com menor custo + alpha * (candidato com menor custo – candidato com maior custo).

Na segunda fase do GRASP, foram utilizadas seqüencialmente as heurísticas de buscas locais: Inserir Atendimentos (para caso de o construtivo não obter sucesso no atendimento a todas as demandas), Eliminar Rotas, Mudar Atendimentos, Trocar Atendimentos e Resequenciar Atendimentos. A ordem em que as buscas locais são aplicadas influi na solução final de cada iteração e conseqüentemente na solução final. Cada nova solução gerada por uma busca local é utilizada como parâmetro de entrada das heurísticas de busca local posterior.

Para as duas versões das heurísticas baseadas no GRASP os melhores resultados foram obtidos com o alpha assumindo os valores:

- 0,05;
- 0,10;
- 0,15;
- 0,20.

6.1. Versão Vizinho mais Próximo

A versão deste algoritmo para a fase de construção do GRASP foi adaptada a partir do algoritmo descrito do Capítulo 4 e de acordo com o modelo do PRV proposto.

O parâmetro “número de iterações” foi fixado em 200 objetivando limitar a execução do GRASP, para o conjunto de instâncias propostas por este trabalho, em aproximadamente 30 minutos, pois o objetivo era obter uma solução dentro de um tempo razoável para prática.

Basicamente o que difere a versão desta heurística para a original é a criação da Lista de Candidatos Restrito e a escolha aleatória a partir desta lista, com aplicação de um filtro se utilizando do valor de alpha, tornado-a assim aleatorizada e não mais simplesmente gulosa.

O pseudocódigo da versão implementada por este trabalho é o seguinte:

01.	CONTRUTIVO vizinho mais próximo;
02.	
03.	ENTRADA: instância;
04.	SAÍDA : instância atualizada + solução viável;
05.	
06.	INÍCIO
07.	INICIAR solução;
08.	ORDERNAR veículos pelo menor custo fixo;
09.	ENQUANTO número de pedidos < total de pedidos E existirem veículos disponíveis FAÇA
10.	INICIAR rota;
11.	atendimento O = depósito;
12.	ENQUANTO existir atendimentos possíveis de serem visitados FAÇA
13.	INICIAR LCR;
14.	PARA todos os atendimentos K não roteados E atendimento K ≠ depósito FAÇA
15.	SE distância(atendimento O, atendimento K) < menor ENTÃO
16.	SE doca(s) do veículo é compatível com doca do atendimento K ENTÃO
17.	SE tempo viagem(atendimento O, atendimento K)
18.	< fechamento janela tempo do atendimento K ENTÃO
19.	SE restrições do veículo são respeitadas ENTÃO
20.	ADICIONAR atendimento k à LCR;
21.	FIM-SE;
22.	FIM-SE;
23.	FIM-SE;
24.	FIM-SE;
25.	FIM-PARA;
26.	SE LCR contiver elementos ENTÃO
27.	FILTRAR a partir do alpha os atendimentos para o sorteio;
28.	SORTEAR atendimento;
29.	ADICIONAR atendimento sortear a rota;
30.	atendimento O = atendimento sorteado;
31.	FIM-SE;
32.	FIM-ENQUANTO;
33.	SE rota possui atendimentos ENTÃO
34.	ADICIONAR rota;
35.	FIM-SE;
36.	FIM-ENQUANTO;
37.	FIM.

Figura 21: Pseudocódigo Construtivo Vizinho mais Próximo

6.2. Versão Economias 2

Esta versão do construtivo do GRASP, assim como a primeira versão da fase de construção, é uma adequação da heurística de Economias versão 2, descrito do Capítulo 4 e de acordo com o modelo do PRV proposto.

O “parâmetro número” de iterações foi fixado em 50, com o mesmo objetivo da primeira versão da metaheurística, limitar o tempo de execução do GRASP, para o conjunto de instâncias propostas por este trabalho, em aproximadamente 30 minutos. O

número de iterações decresceu significativamente, pelo custo computacional da heurística de Economias – versão 2 ser bem superior ao Vizinho mais Próximo.

A diferença entre a versão aleatorizada e a simplesmente gulosa do algoritmo de Economias – versão 2 é a atualização da lista de economias para cada união realizada entre as rotas, e a escolha da economia acontece de maneira aleatória a partir de um filtro baseado no valor do parâmetro alpha e um sorteio.

O pseudocódigo desta versão da fase de construção do GRASP, baseada na heurística de Economias – versão 2, implementada é o seguinte:

01.	CONTRUTIVO economias – versão 2;
02.	
03.	ENTRADA: instância;
04.	SAÍDA : instância atualizada + solução viável;
05.	
06.	INÍCIO
07.	CRIAR SOLUÇÃO INICIAL;
08.	CRIAR LISTA DE ECONOMIAS;
09.	ENQUANTO existirem economias viáveis FAÇA
10.	FILTRAR a partir do alpha as economias da lista de economias para o sorteio;
11.	SORTEAR economia;
12.	UNIR ROTAS;
13.	ATUALIZAR LISTA DE ECONOMIAS;
14.	FIM-PARA;
15.	FIM-ENQUANTO;
16.	ALOCAR VEICULOS;
17.	CALCULAR custo solução;
18.	
19.	FIM.

Figura 22: Pseudocódigo Economias – versão 2

7. Instâncias

As instancias de teste foram criadas a partir de dados reais obtidos junto a uma grande empresa no setor varejista. Estas informações estão estruturadas em um formato similar ao padrão utilizado por um pacote de roteamento empregado no mercado, já detalhado nos Capítulos 2 e 3 deste trabalho.

No total criou-se um conjunto contendo 20 instâncias, agrupadas de duas em duas, de acordo com o número de atendimentos a serem realizados. As duas menores instâncias contêm 65 atendimentos, as duas seguintes 80, e assim por diante, em intervalos de 15 em 15 até chegar a 200 pedidos. As distâncias são calculadas a partir da latitude e longitude, utilizando geometria esférica, entre todos os pares de pontos, clientes e depósito, em questão.

Uma descrição dos dados que formam este estudo de caso é: a instância I para o Problema de Roteamento de Veículos proposto, composta por um conjunto de dados, $I = (L, J, P, O, T, V)$, definida como: L, conjunto de locais com seus respectivos dados referentes à identificação, localização geográfica, determinação da classe de janelas de tempo para a visita de veículos, definição de valores fixos de tempo para carregamento e descarregamento de cargas, e tipo de docagem; J representando os dados referentes a cada classe de janela de tempo onde é descrito o dia da semana, a hora inicial e final que um local pode ser atendido por um veículo; P, relação dos produtos a serem entregue, identificando e determinando valores para conversão da quantidade do produto em uma unidade padrão; O, descrição de pedidos indicando sua origem, a partir do centro de distribuição, para o local de destino; T representa o conjunto de produtos e respectivas quantidades de cada produto para o pedido específico; e V, dados da frota heterogênea de veículos, com a quantificação dos custos, velocidade máxima e capacidade. O conjunto de dados L, J, P, O, T, V estão estruturados em forma de arquivos texto, onde cada arquivo está organizado através de campos. No Apêndice A é mostrado a composição de cada arquivo com uma descrição para cada campo.

A quantidade de registros em cada arquivo texto, que compõem a instância, é detalhada na tabela a seguir com a finalidade de uma maior compreensão do caso em estudo.

Tabela 4: Relação do número de registros para as instâncias

INSTÂNCIA	LOCAIS	JANELAS	PRODUTOS	PEDIDOS	PEDITENS	VEICULOS
065a	65	5	70	64	121	33
065b	65	5	84	64	129	33
080a	80	5	74	79	150	33
080b	80	5	83	79	144	33
095a	95	5	81	94	167	33
095b	95	5	84	94	175	33
110a	110	5	89	109	181	33
110b	110	5	90	109	214	33
125a	125	5	86	124	220	33
125b	125	5	98	124	240	33
140a	140	5	121	139	448	33
140b	140	5	109	139	446	33
155a	155	5	158	154	519	33
155b	155	5	142	154	486	33
170a	170	5	155	169	570	33
170b	170	5	169	169	647	33
185a	185	5	171	184	727	33
185b	185	5	152	184	536	33
200a	200	5	177	199	735	33
200b	200	5	166	199	650	33

É importante destacar que os arquivos de janelas de tempo, veículos e docas se mantêm fixos para todas as instâncias, isto porque se trata de dados reais, onde o conjunto de clientes é pouco dinâmico e a frota de veículos é de propriedade da própria empresa que originou estes dados.

8. Resultados Computacionais

Os resultados computacionais buscam principalmente, neste trabalho, demonstrar experimentos comparativos. Esta modalidade de experimento tem como objetivo fornecer provas da superioridade das idéias propostas, demonstrar as qualidades e fraquezas das estratégias adotadas pelos algoritmos descritos, de forma a manter a parcialidade e descrição objetiva das fatos, coletados e observados durante os experimentos. Importante trabalhos descritivos para desenvolvimento de experimentos são [27], [28] e [29].

Os resultados dos experimentos neste trabalho apresentado são tanto de interesse acadêmico quanto do mercado, visto que as instâncias e métodos utilizados são inerentes a situações reais.

Os testes foram executados em um computador PC AMD Sempron 2400+, 512 MB de RAM. Os algoritmos do Vizinho mais Próximo, Economias, os algoritmos de busca local e as versões GRASP foram implementados na linguagem C. Os testes realizados com o pacote de roteamento também foram executados no mesmo computador.

Para cada uma dessas vinte instâncias encontrou-se: quatorze soluções executando cada um dos algoritmos disponíveis no pacote utilizado na comparação; 14 soluções obtidas pela aplicação das buscas locais sobre as soluções fornecidas pelo pacote; uma solução obtida da Heurística do Vizinho mais Próximo; uma solução obtida da Heurística de Economias versão 1; uma solução obtida da Heurística de Economias versão 2; três solução aplicando-se as buscas locais sobre as heurísticas construtivas; 8 soluções rodando-se o GRASP com parâmetro $\alpha = 0,05, 0,10, 0,15$ e $0,20$, metade para o GRASP baseado no Vizinho mais próximo e a outra parte para o GRASP originário da heurística de Economias versão 2, totalizando 42 versões e 42×20 execuções.

Os dados da próxima tabela representam a seguinte situação. Para cada uma das 20 instâncias de teste, as soluções fornecidas por cada método foram ordenadas de forma que o melhor método nessa instância recebeu classificação 1, o segundo melhor a classificação 2, e assim por diante. A coluna "Soma Rank" apresenta a soma das 20 classificações obtidas por cada método. Quanto menor for esse valor, melhor o desempenho do método em comparação com os demais. A coluna tempo médio é o tempo médio de CPU em segundos.

Observe que para os métodos que envolvem o pacote de roteamento não é apresentado o tempo, pois o pacote não permite a medição do tempo de execução dos seus algoritmos. Estima-se que os tempos médios de execução para cada método fiquem em torno de 5 segundos. O tempo das buscas locais é da mesma ordem de grandeza.

A coloração nas linhas da tabela têm o propósito de identificar as classes de algoritmos a que pertencem os métodos, ou seja, a estratégia adotada para a resolução do problema. O verde representa o GRASP baseado na heurística de Economias versão 2, azul o GRASP baseado na heurística do Vizinho mais Próximo, o cinza os métodos oriundos do pacote de roteamento mais as Buscas Locais, rosa a heurística de Economias versão 2 com e sem as Buscas Locais, laranja a heurística do Vizinho mais Próximo com e sem as Buscas Locais, amarelo a heurística de Economias versão 1 com e sem as Buscas Locais e cinza os métodos do pacote de roteamento.

Tabela 5: Classificação média dos algoritmos nas instâncias

Rank	Métodos	Soma Rank	Tempo Médio (segundos)
1	GRASP - Economias - versão 2 - 0,05	181	861,46
2	GRASP - Economias - versão 2 - 0,10	220	831,88
3	GRASP - Vizinho mais Próximo - 0,05	234	937,51
4	GRASP - Economias - versão 2 - 0,15	237	877,7
5	Savings + Buscas Locais	246	-
6	Nearest Pass + Buscas Locais	270	-
7	GRASP - Vizinho mais Próximo - 0,10	275	790,14
8	Boundary + Buscas Locais	288	-
9	Cluster + Weighted Constraint + Buscas Locais	299	-
10	GRASP - Economias - versão 2 - 0,20	305	905,81
11	Cluster + Pie + Buscas Locais	306	-
12	Cluster + Boundary + Buscas Locais	307	-
13	Cluster + Tree + Buscas Locais	329	-
14	Economias - versão 2 + Buscas Locais	330	31,7
15	Cluster + Nearest Pass + Buscas Locais	334	-
16	Tree + Buscas Locais	335	-
17	Weighted Constraint + Buscas Locais	338	-
18	GRASP - Vizinho mais Próximo - 0,15	339	701,56
19	Vizinho mais Próximo + Buscas Locais	344	5,24
20	Cluster + Savings + Buscas Locais	350	-
21	Cluster + Nearest Neighbor + Buscas Locais	363	-
22	Pie + Buscas Locais	363	-
23	GRASP - Vizinho mais Próximo - 0,20	402	640,76
24	Cluster + Weighted Constraint	424	-

25	Weighted Constraint	432	-
26	Nearest Pass	442	-
27	Economias - versão 1 + Buscas Locais	445	4,47
28	Nearest Neighbor + Buscas Locais	446	-
29	Boundary	466	-
30	Savings	471	-
31	Tree	477	-
32	Economias - versão 2	557	25,57
33	Cluster + Savings	576	-
34	Cluster + Tree	583	-
35	Vizinho mais Próximo	615	0,05
36	Cluster + Pie	684	-
37	Cluster + Nearest Pass	696	-
38	Cluster + Boundary	697	-
39	Pie	720	-
40	Economias - versão 1	750	0,51
41	Nearest Neighbor	779	-
42	Cluster + Nearest Neighbor	805	-

O melhor método segundo essa comparação foi o GRASP (Economia versão 2) com parâmetro α 0,05. Entretanto, os tempos médios de execução dos métodos baseados em GRASP são significativamente maiores do que os demais métodos. Entre os métodos rápidos, o de melhor média foi o Savings + Buscas Locais. O método Vizinho mais Próximo + Buscas Locais e a heurística de Economias – versão 2 + Buscas Locais tiveram ranking melhor do que qualquer método do pacote de roteamento isolado, mas perdem para várias combinações de métodos do Pacote de Roteamento + Buscas Locais.

Uma importante observação realizada durante o período de testes, em relação aos resultados dados pelos algoritmos do pacote de roteamento utilizado nas comparações, é que alguns métodos não obtêm sucesso no roteamento de todos os atendimentos, em determinadas instâncias. Para um aperfeiçoamento destas soluções (inserção dos atendimentos não realizados à solução) foi aplicada a busca local Inserir Atendimentos às soluções dadas pelos métodos que não atenderam todos os pedidos.

Os resultados demonstram, mais uma vez, que o desenvolvimento e aplicação de buscas locais a este determinado pacote de roteamento melhoram as soluções sensivelmente. A tabela a seguir lista as instâncias que não apresentam todos seus pedidos atendidos, a quantidade de atendimentos não realizados e o número de pedidos

atendidos após a execução da heurística Inserir Atendimentos estão relacionadas a cada método e instância.

Tabela 6: Resultados da heurística Inserir Atendimentos

Métodos Pacote de Roteamento	Instâncias																				
	155a			170a			170b			185a			185b			200a			200b		
	R	I	P%	R	I	P%	R	I	P%	R	I	P%	R	I	P	R	I	P	R	I	P
Boundary	-	-	-	-	-	-	01	00	100	-	-	-	-	-	-	06	06	0	02	02	0
Nearest Neighbor	01	00	100	01	01	0	03	01	67	03	03	0	02	01	50	06	03	50	03	02	33
Nearest Pass	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	01	01	0	03	03	0
Pie	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Savings	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	05	05	0	02	01	50
Tree	-	-	-	-	-	-	01	01	0	01	01	0	01	01	0	06	06	0	01	01	0
Weighted Constraint	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	05	05	0	02	01	50
Cluster + Boundary	01	00	100	02	00	100	04	00	100	14	00	100	-	-	-	07	03	57	03	01	67
Cluster + Nearest Neighbor	01	00	100	08	00	100	15	01	93	14	00	100	02	00	100	12	03	75	03	00	100
Cluster + Nearest Pass	01	00	100	01	00	100	02	00	100	14	00	100	-	-	-	09	01	89	03	01	67
Cluster + Pie	-	-	-	-	-	-	-	-	-	14	00	100	-	-	-	02	00	100	-	-	-
Cluster + Savings	01	00	100	-	-	-	01	01	0	01	01	0	-	-	-	05	05	0	03	03	0
Cluster + Tree	-	-	-	-	-	-	01	01	0	01	01	0	-	-	-	05	05	0	03	03	00
Cluster + Weighted Constraint	01	00	100	-	-	-	-	-	-	01	01	0	-	-	-	05	05	0	02	02	00

R = Método Pacote de Roteamento sem busca local Inserir Atendimentos

I = Método Pacote de Roteamento com busca local Inserir Atendimentos

P = Percentagem de melhoria

A Tabela 7 apresenta uma comparação entre três possíveis abordagens que um hipotético cliente poderia utilizar para resolver esse tipo de instâncias. A primeira seria adquirir o pacote (que custa dezenas de milhares dólares por licença) e sempre aplicar todos os 14 algoritmos disponíveis em cada instância, escolhendo a melhor solução

obtida. A segunda seria adquirir o pacote, rodar todos os 14 algoritmos, ler os arquivos de saída e aplicar as buscas locais sobre cada solução obtida. A terceira abordagem seria aplicar apenas os algoritmos apresentados nesse trabalho, sem o uso de nenhum pacote. Para cada uma das abordagens são relacionadas, por instância, custo, desvio da melhor solução encontrada e tempo de execução.

O tempo médio das duas abordagens que envolvem os métodos do pacote de roteamento utilizado não são relacionadas pelo mesmos motivos já descritos anteriormente (tabela 5), porém destaca-se novamente que a média de tempo é da grandeza de 5 segundos tanto para cada um dos métodos do pacote utilizado e das Buscas Locais. Isso totaliza 70 segundos para rodar os 14 métodos e mais cerca de 70 segundos para as buscas locais.

Tabela 7: Visualização tabular das comparação entre abordagens

Instância	Pacote de Roteamento		Pacote de Roteamento + Buscas Locais		Algoritmos Propostos			Melhor Custo
	CUSTO	DESV.	CUSTO	DESV.	CUSTO	DESV.	TEMPO	
065a	R\$ 1.075,62	20,98	R\$ 889,09	0,00	R\$ 917,40	3,18	93,680	R\$ 889,09
065b	R\$ 1.068,38	0,80	R\$ 1.059,88	0,00	R\$ 1.096,53	3,46	136,390	R\$ 1.059,88
080a	R\$ 1.109,87	4,37	R\$ 1.063,39	0,00	R\$ 1.116,89	5,03	104,040	R\$ 1.063,39
080b	R\$ 1.058,94	1,14	R\$ 1.046,99	0,00	R\$ 1.187,50	13,42	285,720	R\$ 1.046,99
095a	R\$ 2.554,29	10,85	R\$ 2.304,28	0,00	R\$ 2.310,11	0,25	1,560	R\$ 2.304,28
095b	R\$ 2.583,93	10,76	R\$ 2.333,01	0,00	R\$ 2.396,66	2,73	280,900	R\$ 2.333,01
110a	R\$ 1.491,21	5,53	R\$ 1.413,03	0,00	R\$ 1.539,12	8,92	439,740	R\$ 1.413,03
110b	R\$ 1.377,18	0,68	R\$ 1.367,83	0,00	R\$ 1.517,81	10,97	491,520	R\$ 1.367,83
125a	R\$ 3.314,88	10,93	R\$ 2.989,69	0,05	R\$ 2.988,16	0,00	508,350	R\$ 2.988,16
125b	R\$ 1.501,43	1,95	R\$ 1.472,68	0,00	R\$ 1.564,74	6,25	6,840	R\$ 1.472,68
140a	R\$ 2.522,33	1,75	R\$ 2.478,88	0,00	R\$ 2.890,41	16,60	1202,090	R\$ 2.478,85
140b	R\$ 2.662,87	8,57	R\$ 2.452,58	0,00	R\$ 2.748,76	12,08	5,900	R\$ 2.452,58
155a	R\$ 5.800,39	10,59	R\$ 5.730,21	9,25	R\$ 5.244,82	0,00	1353,670	R\$ 5.244,82
155b	R\$ 4.887,63	9,53	R\$ 4.736,86	6,15	R\$ 4.462,24	0,00	1336,010	R\$ 4.462,24
170a	R\$ 6.218,91	10,93	R\$ 6.056,73	8,03	R\$ 5.606,32	0,00	36,750	R\$ 5.606,32
170b	R\$ 7.121,29	14,03	R\$ 7.025,16	12,50	R\$ 6.244,85	0,00	1768,047	R\$ 6.244,85
185a	R\$ 7.977,00	3,79	R\$ 7.841,06	2,02	R\$ 7.685,95	0,00	1739,803	R\$ 7.685,95
185b	R\$ 5.594,79	19,01	R\$ 5.016,97	6,72	R\$ 4.700,94	0,00	58,730	R\$ 4.700,94
200a	R\$ 10.077,73	41,34	R\$ 7.655,87	7,38	R\$ 7.129,91	0,00	2131,685	R\$ 7.129,91
200b	R\$ 7.596,03	27,63	R\$ 6.372,31	7,07	R\$ 5.951,43	0,00	117,960	R\$ 5.951,43
MÉDIA SOMA	R\$ 75.010,77	10,76	R\$ 71.306,50	2,96	R\$ 69.300,55	4,14	604,97	R\$ 67.896,26

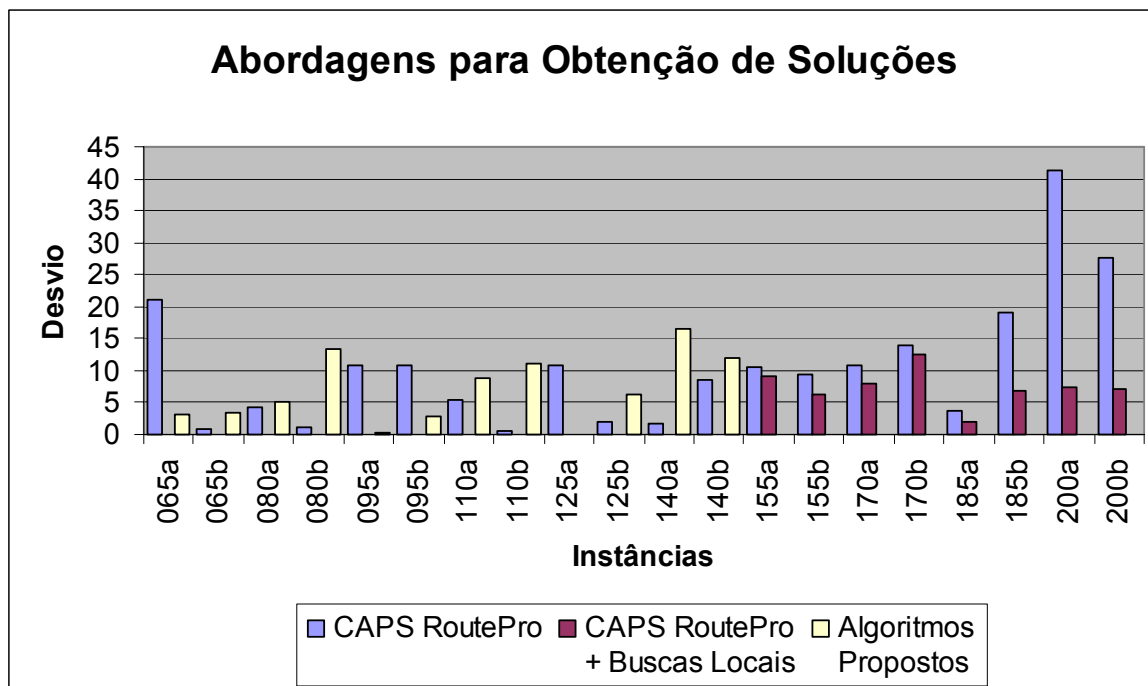


Figura 23: Visualização gráfica das comparação entre abordagens

A melhor abordagem em termos de qualidade média de solução é Pacote de Roteamento + Buscas Locais, com 2,96% de desvio em relação as melhores soluções. Entretanto, a abordagem apenas com os novos algoritmos propostos é claramente melhor para as instâncias maiores. Como o valor absoluto do custo dessas instâncias são mais elevados, a melhor abordagem em termos do custo total para rotear as 20 instâncias acaba sendo melhor com os algoritmos propostos, R\$ 69.300,55 contra R\$ 71.306,50. É ainda mais importante notar que o custo total apenas com os métodos do Pacote de Roteamento é de R\$ 75.010,77, quando sabemos que existem soluções (que quase certamente não são ótimas) com custo de R\$ 67.896,26.

9. Conclusões e Propostas Futuras

A partir dos resultados obtidos podem-se apresentar algumas observações:

- Executar muitos algoritmos construtivos diferentes (no caso todos os 14 métodos do pacote de roteamento utilizado) pode levar a soluções significativamente melhores do que escolher apenas os 2 ou 3 métodos com melhor desempenho médio na classe de instâncias em questão (o que é recomendado na documentação do pacote de roteamento utilizado nas comparações);
- o uso das buscas locais melhora bastante o custo médio das soluções obtidas apenas com o pacote. Esse resultado é bastante significativo do ponto de vista econômico e claramente aponta para a necessidade de que os pacotes, além das heurísticas construtivas, também incorporem algoritmos mais avançados, de eficácia já comprovada pela pesquisa acadêmica;
- a utilização isolada da busca local Inserir Atendimentos já melhora sensivelmente a aplicabilidade prática do pacote de roteamento utilizado nos testes, pois numa situação real não é possível deixar de realizar atendimentos programados. Essa simples busca local realiza em cerca de 1 segundo uma tarefa extremamente penosa de ser realizada manualmente, descobrir em que pontos das rotas já existentes pode-se inserir clientes não atendidos;
- novas heurísticas desenvolvidas, em particular as duas versões baseadas na metaheurística GRASP, para as instâncias maiores apresentam soluções melhores do que os métodos do pacote acrescidos das buscas locais. Isso também é muito importante do ponto de vista econômico. Dado que essas instâncias maiores envolvem maiores custos, ganhos percentuais da ordem de 5 a 10 por cento representam quantias elevadas;

Como proposta futuras e extensões a este trabalho ficam:

- a intensificação da busca nas heurísticas baseadas na técnica do GRASP, através da reconecção por caminhos (Path Relinking);
- a implementação de novas heurísticas construtivas, inclusive para a fase de construção do GRASP;
- o desenvolvimento de novas metaheurísticas para comparação dos resultados obtidos pelo GRASP para este conjunto de instâncias;
- a aplicação de uma nova camada para refinamento das soluções, baseadas em métodos de Programação Inteira;

- uma análise visual dos desenhos que as rotas formam, objetivando uma interpretação das diferenças entre as soluções, principalmente focando o percurso de um veículo;
- buscar a compreensão do por que algumas heurísticas funcionam melhor que as outras em determinadas instâncias, quais as características determinam a construção de uma melhor solução para cada estratégia proposta pelos algoritmos.

Referências Bibliográficas

- [01] E. Baker. Evolution of vehicle routing software: case studies in the United States. Em “The vehicle routing problem” (P. Toth e D. Vigo Eds.), 353-361, 2002.
- [02] Caps Logistics, RoutePro Dispatcher 7.0 Version – User Manual, Atlanta, 2002.
- [03] G. Dantzig, J. Ramser. The truck dispatching Problem. *Management Science*, 6:80-91, 1959.
- [04] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedure. *Journal of Global Optimization*, 6:109-133, 1995.
- [05] G. Laporte e F. Semet. Classical Heuristics for the Capacitated VRP. Em “The vehicle routing problem” (P. Toth e D. Vigo Eds.), 109-128, 2002.
- [06] M. Desroches, J. K. Lenstra, e M. W. P. Savelsbergh. A classification scheme for vehicle routing and scheduling problems. *Journal of Operation Research Society*, 46:322-332, 1990.
- [07] G. Laporte e Y. Nobert. Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics*, 31:147-184, 1987.
- [08] N. Christofides, A. Mingozzi e P. Toth. The vehicle routing problem. Em N. Christofides, A. Mingozzi P. Toth, e C. Sandi, editores. *Combinatorial Optimization*, Wiley, Chichester, UK, 1979, pp, 315-338.
- [09] L. D. Bodin, B. L. Golden, A. A. Assad, e M. Ball. Routing and scheduling of vehicles and crews, the state of the art. *Computers and Operations Reasearch*, 10(2):63-212, 1983.
- [10] N. Cristofides. Vehicle routing. Em E. L. Lawler J. K. Lenstra, A. H. G. Rinnooy Kan, e D. B. Shmoys, editores. *The traveling Salesman Problem*, Wiley, Chichester, UK, 1985, pp. 431-448.

- [11] G. Laporte. The vehicle routing problem. An overview of exact and approximate algorithms. *European Journal of Operation Research*, 59:345-358, 1992.
- [12] P. Toth e D. Vigo. Exact algorithms for vehicles routing. Em T. Crainic and G. Laporte, editores, *Fleet Management and Logistics*, Kluwee, Boston, MA, 1998, pp. 1-31.
- [13] B. L. Golden, E. A. Wasil, J. P. Kelly, e I. M. Chao. Metaheuristics in vehicles routing. Em T. G. Cranic e G. Laporte, editores, *Fleet Management and Logistics*, Kluwer, Boston, MA, 1998, pp. 33-56.
- [14] G. Laporte. Vehicle routing. Em M. Dell’Amico, F. Maffioli, e S. Martello, editores, *Annotated Bibliographies in Combinatorial Optimization*, Wiley, Chichester, UK, 1997.
- [15] G. Laporte e I.H. Osman. Routing problems: a bibliography. *Annals of operations. Annals of Operations Research*, 61:227-262, 1995.
- [16] B. L. Golden e A. A. Assad. *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam, 1998.
- [17] P. Toth e D. Vigo Eds. *The Vehicle Routing Problem*, SIAM Monographs Discrete Mathematics and Applications, 2002.
- [18] M. Gendreau, G. Laporte e J. Y. Potvin. Metaheuristics for the Capacitated VRP. Em “The vehicle routing problem” (P. Toth e D. Vigo Eds.), 129-149, 2002.
- [19] G. Clarke e J. V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:586-581, 1964.
- [20] L. D. Aronson. *Algorithms for vehicle routing – A survey*. Faculty of Technical Mathematics and Informatics, 1995.
- [21] R. G. I. Arakaki. O problema de roteamento de veículos e algumas metaheurísticas. Monografia apresentada para o Exame de Qualificação do Curso de Computação Aplicada – CAP, Instituto Nacional de Pesquisas Espaciais – INPE. Setembro, 1998.

[22] F. Glover, A. G. Kochenberger Eds. HandBook of metaheuristics. Klumer Academic Publishers, Boston/Dordrecht/London, 2003.

[23] L. H. Osman e G. Laporte. Metaheuristics: a bibliography. *Annals of Oper. Res.*, 63:513-623, 1996.

[24] S. Voss, S. Martello, L. H. Osman e C. Roucairol. Metaheuristics: advances and trends in local search paradigms for optimization. Kluwer Academic Publihers, 1999.

[25] C. Blum e A. Roli. Metaheuristics in combinatorial optimization: overview and conceptual comparison. Tech Report – Université Libre de Bruxelles, 2001.

[26] W. Chaovalitwongs, D. Kim e P. Pardalos. GRASP with a New Local Search Scheme for Vehicle Routing Problems with Time Windows. *Journal of Combinatorial Optimization*, 2003;7(2):111-207

[27] C.C. McGeoch e B.M.E. Moret, "How to present a paper on experimental work with algorithms", published in *SIGACT News*, Vol. 30, No. 4, December 1999.

[28] B.L. Barr, J.P. Golden, J. P. Kelly, M.G.C Resend, e W.R. Stewart, Designing and reporting on computacional experiments with heuristic methods, *Journal of Heuristics*, vol. 1, pp. 9-32, 1995.

[29] É. D. Taillard, "A statistical test for comparing rates of success", Extended abstract, Metaheuristic international conference MIC 03, Kyoto, Japan, August 2003.

Apêndice A – Estrutura dos dados nas instâncias

Um fato a se observar nesta descrição é que alguns campos dos arquivos detalhados nas tabelas não são necessários na representação do modelo proposto, sendo os mesmos somente indispensáveis no momento em que um novo modelo for recomendado com estas características, a princípio não requerido neste trabalho.

Estes campos são mencionados com o pretexto de destacar a necessidade obrigatória dos mesmos na execução no pacote de roteamento utilizado, ainda que não resulte em nenhuma alteração na solução do problema em questão, porque os mesmos estão preenchidos com valores nulos.

Tabela 8: Estrutura do arquivo de Locais

L = ARQUIVO DE LOCAIS				
	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ID	String 10	Identificação do local, valor único em todo arquivo.
02	SIM	Type	String	Tipo do local, utilizado para associação de certas características para um conjunto de locais.
03	SIM	Name	String 30	Nome de local.
04	SIM	X	Float	Coordenada correspondente a longitude do local.
05	SIM	Y	Float	Coordenada correspondente a latitude do local.
06	SIM	DockType	String	Tipo de doca de um local, utilizado para determinar o tipo de veículos que pode ser utilizado para entrega ou recolhimento neste local.

07	NÃO	PickWinClass	String 20	Classe que identifica a janela de tempo, quando pode ocorrer, o recolhimento, descrita no arquivo específico.
08	SIM	DropWinClass	String 20	Classe que identifica a janela de tempo, quando pode ocorrer a entrega, descrita no arquivo específico.
09	NÃO	FixLoadTime	Float	Tempo fixo em horas que leva para carregar um veículo no local.
10	SIM	FixUnldTime	Float	Tempo fixo em horas que leva para descarregar um veículo no local.
11	NÃO	City	String 30	Cidade onde se encontra o local.
12	NÃO	State	String	Estado onde se encontra o local.
13	NÃO	EnfDockSched	Boolean	Determina se a restrição de um número de docas será empregada.
14	NÃO	MaxNumDocks	Integer	Número de docas para o local.

Tabela 9: Estrutura do arquivo de Janelas de Tempo

J = ARQUIVO DE JANELAS DE TEMPO				
	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ClassID	String 20	Identificação da classe janela de tempo, valor único em todo arquivo.
02	SIM	Active	String 10	Determina se a classe está ativa ou inativa.

03	NÃO	StarDay	String	Define o dia em que a janela de tempo é aberta para receber veículos, está relacionado com o campo OpenTime.
04	NÃO	CloseDay	String	Define o dia em que a janela de tempo é fechada, não possibilitando a visita de veículos, está relacionado com o campo CloseTime.
05	SIM	OpenTime	String 20	Define a hora em que será aberta a janela de tempo para receber veículos, está relacionada com o campo StarDay.
06	SIM	CloseTime	String 20	Define a hora em que é fechada a janela de tempo, não possibilitando a visita de veículos, está relacionado com o campo CloseDay.
07	NÃO	fMaxWait	String	Quantidade de tempo em que o veículo pode esperar a abertura da janela de tempo.
08	NÃO	waitUnits	String	Unidade de medida para o campo fMaxWait.

Tabela 10: Estrutura do arquivo de Produtos

P = ARQUIVO DE PRODUTOS

	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ID	String 10	Identificação do produto, valor único em todo arquivo.
02	SIM	Description	String 30	Breve descrição do produto.
03	SIM	MultToQty1	Float	Primeiro fator de conversão do produto para uma unidade padrão.
04	NÃO	MultToQty2	Float	Segundo fator de conversão do produto para uma unidade padrão.
05	NÃO	MultToQty3	Float	Terceiro fator de conversão do produto para uma unidade padrão.
06	NÃO	ProdQtyLbl	String	Forma no plural da unidade de medida do produto.
07	NÃO	ProdQtyLbIs	String	Forma no singular da unidade de medida do produto.

Tabela 11: Estrutura do Arquivo de Pedidos

O = ARQUIVO DE PEDIDOS				
	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ID	String 10	Identificação do pedido, valor único em todo arquivo.
02	SIM	Type	String	Tipo do pedido, utilizado para associação de certas características para um

				conjunto de pedidos.
03	SIM	FromID	String	Identificação do local de carregamento do pedido.
04	SIM	ToID	String	Identificação do local de entrega do pedido.

Tabela 12: Estrutura do arquivo de Itens de Pedido

T = ARQUIVO DE ITENS DE PEDIDO				
	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ID	String 10	Identificação do item de pedido, valor único em todo o arquivo.
02	SIM	OrderID	String	Identificação do pedido, presente no arquivo de pedidos, em que este item está relacionado.
03	SIM	ProductType	String	Identificação do produto a que este item de pedido se refere, relacionado com o arquivo de produtos.
04	SIM	ProductQty	Float	Quantidade de unidades pedidas deste produtos.

Tabela 13: Estrutura do arquivo de Veículos

V = ARQUIVO DE VEÍCULOS				
	REQUERIDO	CAMPO	TIPO DE DADO	DESCRIÇÃO
01	SIM	ID	String 10	Identificação do veículo, valor único em todo o arquivo.
02	SIM	Name	String 30	Nome do veículo.
03	NÃO	Type30	String 30	Nome do tipo de

				veículo.
04	NÃO	Priority	Integer	Prioridade em que o algoritmo considera os veículos.
05	SIM	FixedCost	Float (Currency)	Custo fixo por utilizar um veículo em uma rota.
06	NÃO	CostHour	Float (Currency)	Custo por hora para operar o veículo.
07	SIM	CostDistance	Float (Currency)	Custo por distância para utilizar um veículo.
08	NÃO	StopCost	Float (Currency)	Custo associado com cada parada que o veículo faz na rota.
09	SIM	MaxNumRoutes	Integer	Quantidade máxima de rotas que o veículo pode realizar.
10	SIM	MaxHours	Float	Máximo de horas em que um veículo pode realizar uma rota.
11	NÃO	MaxDistance	Float	Distância máxima em que um veículo pode realizar em uma rota.
12	SIM	MaxSpeed	Float	Velocidade máxima que de um veículo.
13	NÃO	MaxTotalDist	Float	Distância máxima de um veículo em todas as rotas.
14	NÃO	MaxTotalHrs	Float	Total de horas em que um veículo pode fazer todas as rotas.
15	NÃO	MaxNumStops	Integer	Número máximo de paradas que um veículo pode realizar

				em uma rota.
16	NÃO	MaxNumOrders	Integer	Número máximo de pedidos que um veículo possa ser carregado.
17	NÃO	MaxNumLoads	Integer	Número máximo de vezes que um veículo pode ser carregado.
18	SIM	MaxCapacity1	Float	Quantidade máxima que um veículo pode transportar de acordo com o primeiro fator de conversão no arquivo produtos.
19	NÃO	MaxCapacity2	Float	Quantidade máxima que um veículo pode transportar de acordo com o segundo fator de conversão no arquivo produtos.
20	NÃO	MaxCapacity3	Float	Quantidade máxima que um veículo pode transportar de acordo com o terceiro fator de conversão no arquivo produtos.
21	SIM	StartID	String	Determina o local que o veículo iniciará a rota.
22	SIM	EndID	String	Determina o local para onde o veículo retornará ao final da rota.
23	NÃO	ReloadPtID	String	Determina o local onde será a recarga com

				produtos caso haja necessidade.
--	--	--	--	------------------------------------

Apêndice B – Resultados individual para cada instância

As tabelas a seguir apresentam uma comparação entre os 42 métodos testados para cada uma das instâncias. A coluna tempo é o tempo médio de CPU em segundos, porém não é possível medir com exatidão os tempos dos métodos pertencentes ao pacote utilizado nas comparações, pois o mesmo não disponibiliza uma opção para a realização da tarefa.

Tabela 14: Resultados instância 065a

TABELAS DE RESULTADOS - instância 065a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Pass + Buscas Locais	2	463,09	R\$ 889,09	0	0,400
02	GRASP - Vizinho mais Próximo - 0,10	2	491,40	R\$ 917,40	0	93,680
03	Tree + Buscas Locais	2	502,01	R\$ 928,01	0	0,130
04	Vizinho mais Próximo + Buscas Locais	2	507,17	R\$ 933,17	0	0,320
05	Cluster + Weighted Constraint + Buscas Locais	3	392,56	R\$ 1.031,56	0	0,98
06	Cluster + Savings + Buscas Locais	3	401,87	R\$ 1.040,87	0	1,360
07	Cluster + Tree + Buscas Locais	3	401,87	R\$ 1.040,87	0	1,370
08	Cluster + Weighted Constraint	3	436,60	R\$ 1.075,62	0	-
09	Nearest Pass	3	444,60	R\$ 1.083,57	0	-
10	Nearest Neighbor + Buscas Locais	3	447,66	R\$ 1.086,66	0	0,510
11	Cluster + Savings	3	448,10	R\$ 1.087,15	0	-
12	Cluster + Tree	3	448,10	R\$ 1.087,15	0	-
13	Savings + Buscas Locais	3	450,84	R\$ 1.089,84	0	0,810
14	GRASP - Economias - versão 2 - 0,05	3	470,87	R\$ 1.109,87	0	57,760
15	GRASP - Economias - versão 2 - 0,15	3	471,72	R\$ 1.110,72	0	52,160
16	Economias - versão 1 + Buscas Locais	3	472,74	R\$ 1.111,74	0	0,950
17	Cluster + Nearest Pass + Buscas Locais	3	473,23	R\$ 1.112,23	0	1,280
18	GRASP - Vizinho mais Próximo - 0,05	3	477,77	R\$ 1.116,77	0	61,500
19	GRASP - Vizinho mais Próximo - 0,20	3	478,80	R\$ 1.117,80	0	105,340
20	Cluster + Pie + Buscas Locais	3	487,97	R\$ 1.126,97	0	0,480
21	GRASP - Economias - versão 2 - 0,20	3	488,15	R\$ 1.127,15	0	59,350
22	GRASP - Vizinho mais Próximo - 0,15	3	497,86	R\$ 1.136,86	0	94,770
23	Boundary + Buscas Locais	3	504,30	R\$ 1.143,30	0	1,020
24	Cluster + Boundary + Buscas Locais	3	506,22	R\$ 1.145,22	0	0,620
25	Cluster + Nearest Neighbor + Buscas Locais	3	507,42	R\$ 1.146,42	0	0,640
26	Tree	3	511,40	R\$ 1.150,37	0	-
27	GRASP - Economias - versão 2 - 0,10	3	512,60	R\$ 1.151,60	0	51,280
28	Weighted Constraint + Buscas Locais	3	514,95	R\$ 1.153,95	0	0,970
29	Weighted Constraint	3	526,40	R\$ 1.165,45	0	-

30	Savings	3	564,20	R\$ 1.203,16	0	-
31	Economias - versão 2 + Buscas Locais	3	620,98	R\$ 1.259,98	0	0,600
32	Economias - versão 2	4	525,35	R\$ 1.377,35	0	0,340
33	Vizinho mais Próximo	4	535,82	R\$ 1.387,82	0	0,000
34	Pie + Buscas Locais	5	485,30	R\$ 1.550,30	0	0,960
35	Cluster + Boundary	5	497,00	R\$ 1.562,87	0	-
36	Cluster + Nearest Pass	5	519,90	R\$ 1.584,87	0	-
37	Boundary	6	449,90	R\$ 1.727,89	0	-
38	Economias - versão 1	5	665,13	R\$ 1.730,13	0	0,030
39	Nearest Neighbor	6	462,30	R\$ 1.740,27	0	-
40	Cluster + Nearest Neighbor	7	524,00	R\$ 2.015,03	0	-
41	Cluster + Pie	6	537,50	R\$ 2.036,48	0	-
42	Pie	10	625,30	R\$ 3.175,09	0	-

Tabela 15: Resultados instância 065b

TABELAS DE RESULTADOS - instância 065b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Pass + Buscas Locais	3	420,88	R\$ 1.059,88	0	0,290
02	Savings + Buscas Locais	3	425,27	R\$ 1.064,27	0	0,870
03	Nearest Pass	3	429,40	R\$ 1.068,38	0	-
04	Weighted Constraint + Buscas Locais	3	430,05	R\$ 1.069,05	0	1,590
05	Cluster + Weighted Constraint + Buscas Locais	3	430,41	R\$ 1.069,41	0	0,620
06	Weighted Constraint	3	436,50	R\$ 1.075,45	0	-
07	Cluster + Weighted Constraint	3	437,10	R\$ 1.076,08	0	-
08	Cluster + Savings + Buscas Locais	3	437,16	R\$ 1.076,16	0	0,270
09	Cluster + Tree + Buscas Locais	3	437,16	R\$ 1.076,16	0	0,270
10	Boundary + Buscas Locais	3	441,21	R\$ 1.080,21	0	1,360
11	Cluster + Savings	3	442,10	R\$ 1.081,09	0	-
12	Cluster + Tree	3	442,10	R\$ 1.081,09	0	-
13	Tree + Buscas Locais	3	444,30	R\$ 1.083,30	0	1,050
14	GRASP - Vizinho mais Próximo - 0,05	3	457,53	R\$ 1.096,53	0	136,390
15	Vizinho mais Próximo + Buscas Locais	3	460,37	R\$ 1.099,37	0	0,500
16	GRASP - Vizinho mais Próximo - 0,15	3	466,30	R\$ 1.105,30	0	66,590
17	GRASP - Vizinho mais Próximo - 0,10	3	478,58	R\$ 1.117,58	0	77,700
18	GRASP - Economias - versão 2 - 0,15	3	482,50	R\$ 1.121,50	0	57,570
19	GRASP - Economias - versão 2 - 0,05	3	494,56	R\$ 1.133,56	0	57,050
20	Tree	3	498,10	R\$ 1.137,10	0	-
21	Cluster + Nearest Pass + Buscas Locais	3	499,21	R\$ 1.138,21	0	0,870
22	Cluster + Boundary + Buscas Locais	3	500,30	R\$ 1.139,30	0	0,800
23	Cluster + Pie + Buscas Locais	3	500,30	R\$ 1.139,30	0	0,810
24	GRASP - Economias - versão 2 - 0,10	3	519,28	R\$ 1.158,28	0	53,780
25	Cluster + Nearest Neighbor + Buscas Locais	3	520,18	R\$ 1.159,18	0	0,740
26	GRASP - Vizinho mais Próximo - 0,20	3	524,41	R\$ 1.163,41	0	67,670
27	GRASP - Economias - versão 2 - 0,20	3	531,46	R\$ 1.170,46	0	53,130
28	Economias - versão 1 + Buscas Locais	3	575,84	R\$ 1.214,84	0	0,800

29	Pie + Buscas Locais	3	659,26	R\$ 1.298,26	0	0,750
30	Savings	4	470,20	R\$ 1.322,17	0	-
31	Nearest Neighbor + Buscas Locais	4	487,95	R\$ 1.339,95	0	1,210
32	Cluster + Nearest Pass	4	501,10	R\$ 1.353,06	0	-
33	Economias - versão 2 + Buscas Locais	4	512,96	R\$ 1.364,96	0	1,320
34	Vizinho mais Próximo	4	578,05	R\$ 1.430,05	0	0,000
35	Cluster + Pie	4	487,80	R\$ 1.486,39	0	-
36	Boundary	5	443,60	R\$ 1.508,58	0	-
37	Cluster + Boundary	5	499,00	R\$ 1.564,00	0	-
38	Economias - versão 2	6	586,79	R\$ 1.864,79	0	0,360
39	Cluster + Nearest Neighbor	7	529,00	R\$ 2.020,01	0	-
40	Economias - versão 1	7	586,35	R\$ 2.077,35	0	0,040
41	Pie	6	542,50	R\$ 2.183,09	0	-
42	Nearest Neighbor	9	504,90	R\$ 2.421,87	0	-

Tabela 16: Resultados instância 080a

TABELAS DE RESULTADOS - instância 080a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Cluster + Weighted Constraint + Buscas Locais	3	424,39	R\$ 1.063,39	0	1,410
02	Nearest Pass + Buscas Locais	3	438,70	R\$ 1.077,70	0	1,800
03	Weighted Constraint + Buscas Locais	3	459,42	R\$ 1.098,42	0	0,620
04	Cluster + Weighted Constraint	3	470,90	R\$ 1.109,87	0	-
05	Savings + Buscas Locais	3	472,01	R\$ 1.111,01	0	1,020
06	GRASP - Economias - versão 2 - 0,10	3	477,89	R\$ 1.116,89	0	104,040
07	Cluster + Savings + Buscas Locais	3	478,87	R\$ 1.117,87	0	1,430
08	Cluster + Tree + Buscas Locais	3	478,87	R\$ 1.117,87	0	1,440
09	GRASP - Economias - versão 2 - 0,05	3	484,26	R\$ 1.123,26	0	103,300
10	Cluster + Pie + Buscas Locais	3	512,98	R\$ 1.151,98	0	1,260
11	GRASP - Vizinho mais Próximo - 0,05	3	513,71	R\$ 1.152,71	0	265,330
12	Cluster + Boundary + Buscas Locais	3	517,61	R\$ 1.156,61	0	1,780
13	Cluster + Nearest Pass + Buscas Locais	3	519,48	R\$ 1.158,48	0	1,540
14	GRASP - Vizinho mais Próximo - 0,15	3	522,73	R\$ 1.161,73	0	208,430
15	GRASP - Vizinho mais Próximo - 0,10	3	526,02	R\$ 1.165,02	0	263,390
16	Vizinho mais Próximo + Buscas Locais	3	534,19	R\$ 1.173,19	0	1,740
17	GRASP - Economias - versão 2 - 0,20	3	536,89	R\$ 1.175,89	0	101,420
18	Tree + Buscas Locais	3	537,54	R\$ 1.176,54	0	1,380
19	GRASP - Economias - versão 2 - 0,15	3	563,08	R\$ 1.202,08	0	95,510
20	GRASP - Vizinho mais Próximo - 0,20	3	575,71	R\$ 1.214,71	0	154,030
21	Boundary + Buscas Locais	3	606,29	R\$ 1.245,29	0	0,930
22	Economias - versão 2 + Buscas Locais	3	622,94	R\$ 1.261,94	0	2,280
23	Pie + Buscas Locais	3	649,05	R\$ 1.288,05	0	1,060
24	Weighted Constraint	4	482,00	R\$ 1.334,02	0	-
25	Cluster + Nearest Neighbor + Buscas Locais	4	491,79	R\$ 1.343,79	0	1,690
26	Economias - versão 1 + Buscas Locais	3	711,84	R\$ 1.350,84	0	0,590
27	Cluster + Nearest Pass	4	500,10	R\$ 1.352,07	0	-
28	Cluster + Savings	4	503,80	R\$ 1.355,78	0	-

29	Cluster + Tree	4	503,80	R\$ 1.355,78	0	-
30	Nearest Neighbor + Buscas Locais	4	516,92	R\$ 1.368,92	0	1,700
31	Savings	4	517,00	R\$ 1.368,96	0	-
32	Nearest Pass	4	537,00	R\$ 1.389,00	0	-
33	Tree	4	576,30	R\$ 1.428,28	0	-
34	Vizinho mais Próximo	4	582,85	R\$ 1.434,85	0	0,010
35	Cluster + Boundary	5	517,10	R\$ 1.582,10	0	-
36	Cluster + Pie	5	519,10	R\$ 1.661,31	0	-
37	Boundary	6	498,00	R\$ 1.776,05	0	-
38	Economias - versão 2	7	577,09	R\$ 2.068,09	0	0,860
39	Nearest Neighbor	8	518,40	R\$ 2.222,42	0	-
40	Economias - versão 1	7	731,84	R\$ 2.222,84	0	0,050
41	Pie	7	568,00	R\$ 2.273,96	0	-
42	Cluster + Nearest Neighbor	9	552,40	R\$ 2.469,39	0	-

Tabela 17: Resultados instância 080b

TABELAS DE RESULTADOS - instância 080b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Tree + Buscas Locais	3	407,99	R\$ 1.046,99	0	0,820
02	Tree	3	419,90	R\$ 1.058,94	0	-
03	Boundary + Buscas Locais	3	429,40	R\$ 1.068,40	0	1,370
04	Cluster + Weighted Constraint + Buscas Locais	3	432,71	R\$ 1.071,71	0	0,520
05	Cluster + Weighted Constraint	3	433,80	R\$ 1.072,75	0	-
06	Cluster + Savings + Buscas Locais	3	448,66	R\$ 1.087,66	0	0,370
07	Cluster + Tree + Buscas Locais	3	448,66	R\$ 1.087,66	0	0,370
08	Nearest Pass + Buscas Locais	3	460,41	R\$ 1.099,41	0	0,380
09	Nearest Pass	3	463,10	R\$ 1.102,10	0	-
10	Savings + Buscas Locais	3	463,64	R\$ 1.102,64	0	0,310
11	Cluster + Pie + Buscas Locais	3	465,45	R\$ 1.104,45	0	0,910
12	Savings	3	468,60	R\$ 1.107,60	0	-
13	Cluster + Nearest Pass + Buscas Locais	3	471,49	R\$ 1.110,49	0	0,600
14	Cluster + Boundary + Buscas Locais	3	494,77	R\$ 1.133,77	0	0,280
15	Weighted Constraint + Buscas Locais	3	547,09	R\$ 1.186,09	0	0,250
16	GRASP - Vizinho mais Próximo - 0,05	3	548,50	R\$ 1.187,50	0	285,720
17	GRASP - Vizinho mais Próximo - 0,10	3	551,90	R\$ 1.190,90	0	246,650
18	GRASP - Economias - versão 2 - 0,05	3	552,09	R\$ 1.191,09	0	104,520
19	GRASP - Vizinho mais Próximo - 0,20	3	561,47	R\$ 1.200,47	0	195,030
20	GRASP - Economias - versão 2 - 0,10	3	578,43	R\$ 1.217,43	0	109,270
21	GRASP - Economias - versão 2 - 0,15	3	578,64	R\$ 1.217,64	0	105,390
22	GRASP - Vizinho mais Próximo - 0,15	3	581,46	R\$ 1.220,46	0	188,550
23	Vizinho mais Próximo + Buscas Locais	3	609,22	R\$ 1.248,22	0	0,730
24	Boundary	4	397,10	R\$ 1.249,12	0	-
25	Nearest Neighbor + Buscas Locais	4	410,95	R\$ 1.262,95	0	0,430
26	Pie + Buscas Locais	4	420,66	R\$ 1.272,66	0	0,440
27	Cluster + Nearest Neighbor + Buscas Locais	4	445,98	R\$ 1.297,98	0	0,510
28	Cluster + Nearest Pass	4	456,50	R\$ 1.308,54	0	-

29	Cluster + Boundary	4	462,50	R\$ 1.314,46	0	-
30	Cluster + Savings	4	489,70	R\$ 1.341,75	0	-
31	Cluster + Tree	4	489,70	R\$ 1.341,75	0	-
32	Weighted Constraint	4	513,20	R\$ 1.365,20	0	-
33	Economias - versão 2 + Buscas Locais	4	573,52	R\$ 1.425,52	0	2,280
34	GRASP - Economias - versão 2 - 0,20	4	615,17	R\$ 1.467,17	0	101,530
35	Economias - versão 1 + Buscas Locais	4	782,94	R\$ 1.634,94	0	0,720
36	Pie	5	446,90	R\$ 1.637,04	0	-
37	Cluster + Pie	5	463,20	R\$ 1.666,00	0	-
38	Vizinho mais Próximo	5	639,32	R\$ 1.704,32	0	0,010
39	Nearest Neighbor	6	446,10	R\$ 1.724,06	0	-
40	Economias - versão 2	6	655,49	R\$ 1.933,49	0	0,650
41	Cluster + Nearest Neighbor	7	483,60	R\$ 1.974,61	0	-
42	Economias - versão 1	6	809,42	R\$ 2.087,42	0	0,050

Tabela 18: Resultados instância 095a

TABELAS DE RESULTADOS - instância 095a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Cluster + Pie + Buscas Locais	7	697,79	R\$ 2.304,28	0	1,070
02	Economias - versão 1 + Buscas Locais	6	851,86	R\$ 2.310,11	0	1,560
03	Boundary + Buscas Locais	8	614,96	R\$ 2.318,96	0	2,260
04	GRASP - Economias - versão 2 - 0,15	7	691,99	R\$ 2.325,39	0	190,410
05	Economias - versão 2 + Buscas Locais	7	702,78	R\$ 2.332,68	0	4,050
06	Pie + Buscas Locais	6	842,33	R\$ 2.345,06	0	0,750
07	GRASP - Economias - versão 2 - 0,05	6	895,59	R\$ 2.347,71	0	218,610
08	GRASP - Economias - versão 2 - 0,10	6	870,98	R\$ 2.350,15	0	209,790
09	GRASP - Economias - versão 2 - 0,20	8	683,01	R\$ 2.387,01	0	184,080
10	GRASP - Vizinho mais Próximo - 0,10	8	728,27	R\$ 2.432,27	0	274,910
11	GRASP - Vizinho mais Próximo - 0,20	8	733,23	R\$ 2.437,23	0	267,170
12	GRASP - Vizinho mais Próximo - 0,05	8	736,02	R\$ 2.440,02	0	304,830
13	GRASP - Vizinho mais Próximo - 0,15	8	749,77	R\$ 2.453,77	0	270,000
14	Nearest Neighbor + Buscas Locais	9	628,90	R\$ 2.545,90	0	2,510
15	Boundary	9	637,30	R\$ 2.554,29	0	-
16	Cluster + Nearest Pass + Buscas Locais	9	645,24	R\$ 2.562,24	0	1,300
17	Nearest Pass + Buscas Locais	9	650,55	R\$ 2.567,55	0	1,340
18	Cluster + Savings + Buscas Locais	9	653,17	R\$ 2.570,17	0	1,220
19	Cluster + Tree + Buscas Locais	9	653,17	R\$ 2.570,17	0	1,220
20	Cluster + Weighted Constraint + Buscas Locais	9	656,30	R\$ 2.573,30	0	1,180
21	Cluster + Boundary + Buscas Locais	9	656,38	R\$ 2.573,38	0	2,420
22	Cluster + Nearest Neighbor + Buscas Locais	9	664,26	R\$ 2.581,26	0	1,530
23	Tree + Buscas Locais	9	679,57	R\$ 2.596,57	0	1,730
24	Savings + Buscas Locais	9	706,43	R\$ 2.623,43	0	2,010
25	Cluster + Weighted Constraint	9	720,50	R\$ 2.637,51	0	-
26	Cluster + Savings	9	721,20	R\$ 2.638,22	0	-
27	Cluster + Tree	9	721,20	R\$ 2.638,22	0	-
28	Cluster + Nearest Pass	9	725,50	R\$ 2.642,47	0	-

29	Vizinho mais Próximo + Buscas Locais	9	734,11	R\$ 2.651,11	0	1,780
30	Cluster + Boundary	9	750,30	R\$ 2.667,34	0	-
31	Weighted Constraint + Buscas Locais	9	753,92	R\$ 2.670,92	0	2,530
32	Tree	9	790,10	R\$ 2.707,07	0	-
33	Weighted Constraint	9	801,50	R\$ 2.718,46	0	-
34	Savings	9	802,70	R\$ 2.719,72	0	-
35	Economias - versão 1	9	727,62	R\$ 2.764,56	0	0,170
36	Nearest Pass	9	845,70	R\$ 2.767,71	0	-
37	Vizinho mais Próximo	9	907,37	R\$ 2.824,37	0	0,010
38	Cluster + Pie	9	772,50	R\$ 2.825,62	0	-
39	Economias - versão 2	10	737,41	R\$ 2.985,27	0	2,140
40	Nearest Neighbor	11	702,70	R\$ 3.045,66	0	-
41	Cluster + Nearest Neighbor	12	754,30	R\$ 3.310,30	0	-
42	Pie	10	755,80	R\$ 3.330,41	0	-

Tabela 19: Resultados instância 095b

TABELAS DE RESULTADOS - instância 095b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Neighbor + Buscas Locais	8	629,01	R\$ 2.333,01	0	1,320
02	GRASP - Vizinho mais Próximo - 0,10	8	692,66	R\$ 2.396,66	0	280,900
03	GRASP - Economias - versão 2 - 0,05	7	736,83	R\$ 2.401,11	0	178,090
04	GRASP - Economias - versão 2 - 0,10	8	709,76	R\$ 2.413,76	0	180,630
05	GRASP - Economias - versão 2 - 0,20	8	729,84	R\$ 2.433,84	0	165,550
06	GRASP - Vizinho mais Próximo - 0,20	8	737,01	R\$ 2.441,01	0	225,330
07	GRASP - Vizinho mais Próximo - 0,05	8	742,41	R\$ 2.446,41	0	280,980
08	GRASP - Vizinho mais Próximo - 0,15	8	753,07	R\$ 2.457,07	0	254,430
09	GRASP - Economias - versão 2 - 0,15	8	762,14	R\$ 2.466,14	0	172,540
10	Economias - versão 2 + Buscas Locais	8	666,40	R\$ 2.506,48	0	3,560
11	Cluster + Boundary + Buscas Locais	9	614,85	R\$ 2.531,85	0	2,460
12	Cluster + Nearest Neighbor + Buscas Locais	9	652,52	R\$ 2.569,52	0	1,610
13	Vizinho mais Próximo + Buscas Locais	8	877,53	R\$ 2.581,53	0	0,810
14	Cluster + Boundary	9	666,90	R\$ 2.583,93	0	-
15	Cluster + Weighted Constraint + Buscas Locais	9	673,98	R\$ 2.590,98	0	2,220
16	Economias - versão 1 + Buscas Locais	9	699,83	R\$ 2.616,83	0	2,360
17	Cluster + Pie + Buscas Locais	5	638,18	R\$ 2.635,85	0	0,590
18	Cluster + Weighted Constraint	9	749,80	R\$ 2.666,76	0	-
19	Weighted Constraint + Buscas Locais	9	750,62	R\$ 2.667,62	0	2,230
20	Cluster + Pie	5	613,20	R\$ 2.690,38	0	-
21	Weighted Constraint	9	788,70	R\$ 2.705,66	0	-
22	Economias - versão 2	9	681,39	R\$ 2.719,51	0	1,800
23	Boundary + Buscas Locais	10	625,88	R\$ 2.755,88	0	2,000
24	Nearest Pass + Buscas Locais	10	636,35	R\$ 2.766,35	0	2,120
25	Cluster + Nearest Pass + Buscas Locais	10	660,88	R\$ 2.790,88	0	2,110
26	Boundary	10	664,90	R\$ 2.794,93	0	-
27	Savings + Buscas Locais	10	671,13	R\$ 2.801,13	0	2,120
28	Tree + Buscas Locais	10	671,44	R\$ 2.801,44	0	2,560

29	Cluster + Savings + Buscas Locais	10	702,43	R\$ 2.832,43	0	2,290
30	Tree	10	720,60	R\$ 2.850,56	0	-
31	Cluster + Nearest Pass	10	722,70	R\$ 2.852,67	0	-
32	Cluster + Tree + Buscas Locais	10	725,63	R\$ 2.855,63	0	2,090
33	Vizinho mais Próximo	9	960,92	R\$ 2.877,92	0	0,020
34	Cluster + Savings	10	759,00	R\$ 2.888,97	0	-
35	Cluster + Tree	10	769,80	R\$ 2.899,79	0	-
36	Nearest Pass	10	788,60	R\$ 2.918,63	0	-
37	Savings	10	806,10	R\$ 2.936,10	0	-
38	Economias - versão 1	10	900,31	R\$ 3.030,31	0	0,100
39	Pie + Buscas Locais	6	552,93	R\$ 3.118,17	0	0,750
40	Pie	6	572,00	R\$ 3.160,56	0	-
41	Nearest Neighbor	12	673,70	R\$ 3.229,67	0	-
42	Cluster + Nearest Neighbor	12	712,30	R\$ 3.268,27	0	-

Tabela 20: Resultados instância 110a

TABELAS DE RESULTADOS - instância 110a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Pass + Buscas Locais	4	561,03	R\$ 1.413,03	0	1,820
02	Cluster + Weighted Constraint + Buscas Locais	4	564,69	R\$ 1.416,69	0	2,760
03	Tree + Buscas Locais	4	576,66	R\$ 1.428,66	0	1,390
04	Nearest Neighbor + Buscas Locais	4	619,59	R\$ 1.471,59	0	1,950
05	Boundary + Buscas Locais	4	630,67	R\$ 1.482,67	0	4,910
06	Tree	4	639,20	R\$ 1.491,21	0	-
07	Cluster + Savings + Buscas Locais	4	643,03	R\$ 1.495,03	0	3,880
08	Cluster + Tree + Buscas Locais	4	643,03	R\$ 1.495,03	0	3,890
09	Savings + Buscas Locais	4	651,96	R\$ 1.503,96	0	3,810
10	Cluster + Nearest Pass + Buscas Locais	4	659,20	R\$ 1.511,20	0	3,880
11	Weighted Constraint + Buscas Locais	4	668,38	R\$ 1.520,38	0	3,030
12	Cluster + Boundary + Buscas Locais	4	672,33	R\$ 1.524,33	0	2,500
13	GRASP - Vizinho mais Próximo - 0,05	4	687,12	R\$ 1.539,12	0	439,740
14	GRASP - Vizinho mais Próximo - 0,10	4	694,13	R\$ 1.546,13	0	447,980
15	Weighted Constraint	4	697,70	R\$ 1.549,67	0	-
16	Cluster + Nearest Neighbor + Buscas Locais	4	706,12	R\$ 1.558,12	0	2,000
17	Vizinho mais Próximo + Buscas Locais	4	713,69	R\$ 1.565,69	0	0,830
18	GRASP - Vizinho mais Próximo - 0,15	4	746,35	R\$ 1.598,35	0	394,830
19	Cluster + Pie + Buscas Locais	4	748,07	R\$ 1.600,07	0	2,550
20	GRASP - Economias - versão 2 - 0,05	4	790,08	R\$ 1.642,08	0	338,810
21	GRASP - Vizinho mais Próximo - 0,20	4	799,63	R\$ 1.651,63	0	384,490
22	Cluster + Weighted Constraint	5	614,10	R\$ 1.679,14	0	-
23	Nearest Pass	5	626,20	R\$ 1.691,27	0	-
24	Cluster + Savings	5	677,70	R\$ 1.742,74	0	-
25	Cluster + Tree	5	677,70	R\$ 1.742,74	0	-
26	GRASP - Economias - versão 2 - 0,10	5	678,94	R\$ 1.743,94	0	318,470
27	GRASP - Economias - versão 2 - 0,15	5	715,30	R\$ 1.780,30	0	306,650
28	Savings	5	729,30	R\$ 1.794,26	0	-

29	Boundary	6	560,70	R\$ 1.838,70	0	-
30	Economias - versão 2 + Buscas Locais	5	781,80	R\$ 1.846,80	0	5,660
31	Pie + Buscas Locais	5	641,30	R\$ 1.850,45	0	5,030
32	GRASP - Economias - versão 2 - 0,20	5	816,59	R\$ 1.881,59	0	299,850
33	Economias - versão 1 + Buscas Locais	5	852,58	R\$ 1.917,58	0	2,530
34	Cluster + Nearest Pass	6	654,40	R\$ 1.932,41	0	-
35	Vizinho mais Próximo	6	773,63	R\$ 2.051,63	0	0,010
36	Cluster + Pie	6	673,20	R\$ 2.063,20	0	-
37	Cluster + Boundary	7	660,40	R\$ 2.151,39	0	-
38	Pie	7	656,20	R\$ 2.510,56	0	-
39	Nearest Neighbor	9	653,40	R\$ 2.570,39	0	-
40	Cluster + Nearest Neighbor	9	685,80	R\$ 2.602,77	0	-
41	Economias - versão 2	10	705,35	R\$ 2.835,35	0	2,980
42	Economias - versão 1	11	977,57	R\$ 3.320,57	0	0,140

Tabela 21: Resultados instância 110b

TABELAS DE RESULTADOS - instância 110b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Pass + Buscas Locais	4	515,83	R\$ 1.367,83	0	2,110
02	Nearest Pass	4	525,20	R\$ 1.377,18	0	-
03	Cluster + Boundary + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,660
04	Cluster + Nearest Neighbor + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,690
05	Cluster + Nearest Pass + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,670
06	Cluster + Pie + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,680
07	Cluster + Savings + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,670
08	Cluster + Tree + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,660
09	Cluster + Weighted Constraint + Buscas Locais	4	529,34	R\$ 1.381,34	0	2,660
10	Tree + Buscas Locais	4	608,70	R\$ 1.460,70	0	3,320
11	Weighted Constraint + Buscas Locais	4	626,95	R\$ 1.478,95	0	3,260
12	Nearest Neighbor + Buscas Locais	4	647,09	R\$ 1.499,09	0	3,100
13	Weighted Constraint	4	648,30	R\$ 1.500,30	0	-
14	GRASP - Vizinho mais Próximo - 0,05	4	665,81	R\$ 1.517,81	0	491,520
15	Vizinho mais Próximo + Buscas Locais	4	672,15	R\$ 1.524,15	0	2,350
16	Savings + Buscas Locais	4	684,47	R\$ 1.536,47	0	3,570
17	GRASP - Vizinho mais Próximo - 0,10	4	699,19	R\$ 1.551,19	0	436,340
18	GRASP - Vizinho mais Próximo - 0,15	4	703,11	R\$ 1.555,11	0	475,820
19	Tree	4	713,30	R\$ 1.565,31	0	-
20	GRASP - Vizinho mais Próximo - 0,20	4	774,87	R\$ 1.626,87	0	411,330
21	GRASP - Economias - versão 2 - 0,05	5	632,56	R\$ 1.697,56	0	344,850
22	Boundary + Buscas Locais	5	638,82	R\$ 1.703,82	0	4,880
23	GRASP - Economias - versão 2 - 0,10	5	667,52	R\$ 1.732,52	0	334,530
24	GRASP - Economias - versão 2 - 0,20	5	678,66	R\$ 1.743,66	0	297,710
25	GRASP - Economias - versão 2 - 0,15	5	725,01	R\$ 1.790,01	0	303,250
26	Economias - versão 2 + Buscas Locais	5	733,86	R\$ 1.798,86	0	8,270
27	Savings	5	775,50	R\$ 1.840,51	0	-
28	Cluster + Savings	6	565,90	R\$ 1.843,86	0	-

29	Cluster + Tree	6	565,90	R\$ 1.843,86	0	-
30	Cluster + Weighted Constraint	6	565,90	R\$ 1.843,86	0	-
31	Cluster + Boundary	6	571,10	R\$ 1.849,06	0	-
32	Cluster + Nearest Pass	6	571,10	R\$ 1.849,06	0	-
33	Cluster + Pie	6	571,10	R\$ 1.921,30	0	-
34	Pie + Buscas Locais	5	764,43	R\$ 1.946,56	0	2,890
35	Economias - versão 1 + Buscas Locais	5	912,37	R\$ 1.977,37	0	1,680
36	Vizinho mais Próximo	6	713,07	R\$ 1.991,07	0	0,020
37	Cluster + Nearest Neighbor	7	578,00	R\$ 2.068,98	0	-
38	Boundary	7	617,30	R\$ 2.108,34	0	-
39	Nearest Neighbor	8	595,60	R\$ 2.299,56	0	-
40	Economias - versão 2	8	762,82	R\$ 2.466,82	0	2,880
41	Pie	10	759,20	R\$ 3.358,15	0	-
42	Economias - versão 1	11	1.029,69	R\$ 3.372,69	0	0,120

Tabela 22: Resultados instância 125a

TABELAS DE RESULTADOS - instância 125a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,05	9	926,67	R\$ 2.988,16	0	508,350
02	Pie + Buscas Locais	9	858,96	R\$ 2.989,69	0	4,890
03	GRASP - Economias - versão 2 - 0,10	9	935,58	R\$ 2.991,64	0	484,700
04	Cluster + Pie + Buscas Locais	10	899,70	R\$ 3.141,93	0	3,670
05	GRASP - Economias - versão 2 - 0,15	9	1.042,62	R\$ 3.163,61	0	458,990
06	GRASP - Vizinho mais Próximo - 0,05	11	896,23	R\$ 3.239,23	0	706,930
07	Nearest Pass + Buscas Locais	11	901,26	R\$ 3.244,26	0	3,330
08	Savings + Buscas Locais	12	692,09	R\$ 3.248,09	0	5,620
09	GRASP - Economias - versão 2 - 0,20	9	1.128,48	R\$ 3.252,41	0	453,430
10	Boundary + Buscas Locais	12	707,36	R\$ 3.263,36	0	6,870
11	Vizinho mais Próximo + Buscas Locais	11	927,85	R\$ 3.270,85	0	3,440
12	Economias - versão 1 + Buscas Locais	11	928,38	R\$ 3.271,38	0	3,300
13	GRASP - Vizinho mais Próximo - 0,15	11	938,25	R\$ 3.281,25	0	601,110
14	Nearest Neighbor + Buscas Locais	12	725,95	R\$ 3.281,95	0	6,630
15	GRASP - Vizinho mais Próximo - 0,10	11	939,54	R\$ 3.282,54	0	662,330
16	Boundary	12	758,90	R\$ 3.314,88	0	-
17	GRASP - Vizinho mais Próximo - 0,20	11	983,79	R\$ 3.326,79	0	543,100
18	Savings	12	774,80	R\$ 3.330,78	0	-
19	Cluster + Tree + Buscas Locais	12	779,69	R\$ 3.335,69	0	5,880
20	Cluster + Weighted Constraint + Buscas Locais	12	783,69	R\$ 3.339,69	0	5,720
21	Cluster + Nearest Pass + Buscas Locais	12	791,13	R\$ 3.347,13	0	5,510
22	Cluster + Boundary + Buscas Locais	12	795,46	R\$ 3.351,46	0	5,720
23	Tree + Buscas Locais	12	800,73	R\$ 3.356,73	0	4,610
24	Cluster + Nearest Neighbor + Buscas Locais	12	800,80	R\$ 3.356,80	0	5,520
25	Cluster + Savings + Buscas Locais	12	800,80	R\$ 3.356,80	0	5,510
26	Economias - versão 2 + Buscas Locais	12	827,01	R\$ 3.383,01	0	8,760
27	Cluster + Weighted Constraint	12	851,20	R\$ 3.407,21	0	-
28	Nearest Pass	11	1.077,30	R\$ 3.420,25	0	-

29	Weighted Constraint + Buscas Locais	12	877,02	R\$ 3.433,02	0	4,050
30	Cluster + Tree	12	878,20	R\$ 3.434,25	0	-
31	Weighted Constraint	12	908,90	R\$ 3.464,94	0	-
32	Vizinho mais Próximo	11	1.129,46	R\$ 3.472,46	0	0,030
33	Tree	12	920,00	R\$ 3.476,45	0	-
34	Cluster + Pie	12	868,40	R\$ 3.521,07	0	-
35	Nearest Neighbor	13	821,90	R\$ 3.590,90	0	-
36	Cluster + Savings	13	87,90	R\$ 3.640,89	0	-
37	Cluster + Nearest Neighbor	13	871,90	R\$ 3.640,89	0	-
38	Cluster + Boundary	13	876,80	R\$ 3.645,77	0	-
39	Cluster + Nearest Pass	13	892,30	R\$ 3.661,29	0	-
40	Economias - versão 2	13	970,68	R\$ 3.739,68	0	5,430
41	Pie	12	878,40	R\$ 3.778,60	0	-
42	Economias - versão 1	13	1.176,85	R\$ 3.945,85	0	0,220

Tabela 23: Resultados instância 125b

TABELAS DE RESULTADOS - instância 125b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Nearest Pass + Buscas Locais	4	620,68	R\$ 1.472,68	0	5,060
02	Nearest Pass	4	649,40	R\$ 1.501,43	0	-
03	Vizinho mais Próximo + Buscas Locais	4	712,74	R\$ 1.564,74	0	6,840
04	Savings + Buscas Locais	4	747,56	R\$ 1.599,56	0	6,360
05	GRASP - Vizinho mais Próximo - 0,05	4	768,83	R\$ 1.620,83	0	774,470
06	GRASP - Vizinho mais Próximo - 0,10	4	831,58	R\$ 1.683,58	0	670,350
07	Tree + Buscas Locais	5	669,94	R\$ 1.734,94	0	7,250
08	GRASP - Vizinho mais Próximo - 0,15	4	883,41	R\$ 1.735,41	0	521,040
09	Cluster + Weighted Constraint + Buscas Locais	5	672,65	R\$ 1.737,65	0	5,670
10	Cluster + Boundary + Buscas Locais	5	719,28	R\$ 1.784,28	0	3,820
11	Cluster + Nearest Neighbor + Buscas Locais	5	719,28	R\$ 1.784,28	0	3,830
12	Cluster + Pie + Buscas Locais	5	719,28	R\$ 1.784,28	0	3,820
13	Boundary + Buscas Locais	5	729,67	R\$ 1.794,67	0	6,650
14	Weighted Constraint + Buscas Locais	5	736,05	R\$ 1.801,05	0	4,370
15	Tree	5	742,90	R\$ 1.807,95	0	-
16	Cluster + Savings + Buscas Locais	5	759,26	R\$ 1.824,26	0	4,830
17	Cluster + Tree + Buscas Locais	5	759,26	R\$ 1.824,26	0	4,820
18	Cluster + Nearest Pass + Buscas Locais	5	767,13	R\$ 1.832,13	0	3,370
19	Nearest Neighbor + Buscas Locais	5	770,80	R\$ 1.835,80	0	3,470
20	Weighted Constraint	5	777,10	R\$ 1.842,14	0	-
21	GRASP - Vizinho mais Próximo - 0,20	4	991,86	R\$ 1.843,86	0	557,000
22	Savings	5	827,00	R\$ 1.892,02	0	-
23	GRASP - Economias - versão 2 - 0,10	5	831,96	R\$ 1.896,96	0	547,220
24	GRASP - Economias - versão 2 - 0,05	5	850,06	R\$ 1.915,06	0	554,560
25	GRASP - Economias - versão 2 - 0,15	5	911,11	R\$ 1.976,11	0	529,610
26	Cluster + Weighted Constraint	6	750,10	R\$ 2.028,05	0	-
27	Cluster + Savings	6	807,30	R\$ 2.085,33	0	-
28	Cluster + Tree	6	807,30	R\$ 2.085,33	0	-

29	Pie + Buscas Locais	5	874,26	R\$ 2.090,98	0	4,290
30	Vizinho mais Próximo	6	837,51	R\$ 2.115,51	0	0,020
31	GRASP - Economias - versão 2 - 0,20	6	862,95	R\$ 2.140,95	0	508,810
32	Economias - versão 2 + Buscas Locais	6	884,25	R\$ 2.162,25	0	10,720
33	Boundary	7	684,50	R\$ 2.175,47	0	-
34	Economias - versão 1 + Buscas Locais	6	961,62	R\$ 2.239,62	0	3,780
35	Cluster + Boundary	8	829,60	R\$ 2.533,65	0	-
36	Cluster + Nearest Pass	8	830,50	R\$ 2.534,52	0	-
37	Cluster + Pie	8	829,60	R\$ 2.669,24	0	-
38	Cluster + Nearest Neighbor	9	845,10	R\$ 2.762,08	0	-
39	Economias - versão 2	10	865,53	R\$ 2.995,53	0	6,160
40	Nearest Neighbor	11	756,60	R\$ 3.099,59	0	-
41	Pie	10	857,90	R\$ 3.685,86	0	-
42	Economias - versão 1	12	1.215,78	R\$ 3.771,78	0	0,180

Tabela 24: Resultados instância 140a

TABELAS DE RESULTADOS - instância 140a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Tree + Buscas Locais	6	1.200,88	R\$ 2.478,88	0	3,940
02	Tree	6	1.244,30	R\$ 2.522,33	0	-
03	Savings + Buscas Locais	6	1.329,60	R\$ 2.607,60	0	5,160
04	Boundary + Buscas Locais	7	1.321,91	R\$ 2.812,91	0	6,630
05	Nearest Pass + Buscas Locais	7	1.359,00	R\$ 2.850,00	0	4,610
06	Cluster + Weighted Constraint + Buscas Locais	7	1.359,93	R\$ 2.850,93	0	7,120
07	Savings	7	1.388,20	R\$ 2.879,17	0	-
08	GRASP - Vizinho mais Próximo - 0,05	6	1.612,41	R\$ 2.890,41	0	1202,090
09	Nearest Pass	7	1.428,90	R\$ 2.919,95	0	-
10	Weighted Constraint + Buscas Locais	7	1.443,01	R\$ 2.934,01	0	5,610
11	Weighted Constraint	7	1.459,10	R\$ 2.950,07	0	-
12	GRASP - Economias - versão 2 - 0,10	7	1.557,18	R\$ 3.048,18	0	862,880
13	GRASP - Economias - versão 2 - 0,20	7	1.573,78	R\$ 3.064,78	0	806,640
14	Nearest Neighbor + Buscas Locais	8	1.369,81	R\$ 3.073,81	0	10,020
15	Cluster + Savings + Buscas Locais	8	1.375,32	R\$ 3.079,32	0	10,940
16	Cluster + Tree + Buscas Locais	8	1.375,32	R\$ 3.079,32	0	10,940
17	GRASP - Vizinho mais Próximo - 0,10	7	1.588,77	R\$ 3.079,77	0	922,250
18	GRASP - Economias - versão 2 - 0,15	8	1.382,02	R\$ 3.086,02	0	792,070
19	Cluster + Weighted Constraint	8	1.395,20	R\$ 3.099,24	0	-
20	Boundary	9	1.196,70	R\$ 3.113,66	0	-
21	Cluster + Boundary + Buscas Locais	8	1.476,33	R\$ 3.180,33	0	5,370
22	GRASP - Economias - versão 2 - 0,05	7	1.374,53	R\$ 3.187,52	0	900,820
23	Economias - versão 1 + Buscas Locais	8	1.538,08	R\$ 3.242,08	0	3,570
24	GRASP - Vizinho mais Próximo - 0,15	7	1.751,38	R\$ 3.242,38	0	775,070
25	Cluster + Savings	8	1.568,50	R\$ 3.272,46	0	-
26	Cluster + Tree	8	1.568,50	R\$ 3.272,46	0	-
27	Cluster + Nearest Pass + Buscas Locais	8	1.569,40	R\$ 3.273,40	0	4,040
28	Economias - versão 2 + Buscas Locais	8	1.285,72	R\$ 3.307,57	0	14,690

29	Cluster + Nearest Neighbor + Buscas Locais	8	1.610,98	R\$ 3.314,98	0	7,130
30	Vizinho mais Próximo + Buscas Locais	7	1.839,98	R\$ 3.330,98	0	10,870
31	GRASP - Vizinho mais Próximo - 0,20	8	1.762,18	R\$ 3.466,18	0	700,820
32	Cluster + Pie + Buscas Locais	7	1.668,69	R\$ 3.469,85	0	2,080
33	Cluster + Nearest Pass	10	1.518,20	R\$ 3.648,17	0	-
34	Pie + Buscas Locais	7	1.651,52	R\$ 3.676,54	0	6,670
35	Vizinho mais Próximo	9	1.949,87	R\$ 3.866,87	0	0,040
36	Economias - versão 1	11	1.599,55	R\$ 3.942,55	0	0,300
37	Nearest Neighbor	12	1.423,00	R\$ 3.978,98	0	-
38	Economias - versão 2	11	1.335,66	R\$ 4.000,86	0	8,790
39	Cluster + Boundary	12	1.550,60	R\$ 4.106,59	0	-
40	Cluster + Nearest Neighbor	14	1.584,00	R\$ 4.566,02	0	-
41	Pie	12	1.571,70	R\$ 5.859,86	0	-
42	Cluster + Pie	13	1.801,60	R\$ 5.911,16	0	-

Tabela 25: Resultados instância 140b

TABELAS DE RESULTADOS - instância 140b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Tree + Buscas Locais	6	1.174,58	R\$ 2.452,58	0	5,310
02	Savings + Buscas Locais	6	1.253,05	R\$ 2.531,05	0	11,090
03	Savings	6	1.384,90	R\$ 2.662,87	0	-
04	Weighted Constraint + Buscas Locais	6	1.398,99	R\$ 2.676,99	0	5,720
05	Boundary + Buscas Locais	7	1.210,05	R\$ 2.701,05	0	15,610
06	Vizinho mais Próximo + Buscas Locais	6	1.470,76	R\$ 2.748,76	0	5,900
07	Tree	7	1.296,10	R\$ 2.787,12	0	-
08	Cluster + Nearest Pass + Buscas Locais	7	1.313,16	R\$ 2.804,16	0	11,270
09	Cluster + Weighted Constraint + Buscas Locais	7	1.324,36	R\$ 2.815,36	0	5,710
10	GRASP - Vizinho mais Próximo - 0,05	6	1.558,49	R\$ 2.836,49	0	1265,430
11	GRASP - Economias - versão 2 - 0,05	7	1.401,34	R\$ 2.892,34	0	950,340
12	Boundary	8	1.196,80	R\$ 2.900,85	0	-
13	GRASP - Economias - versão 2 - 0,15	7	1.421,68	R\$ 2.912,68	0	929,520
14	Weighted Constraint	7	1.429,00	R\$ 2.920,01	0	-
15	Cluster + Savings + Buscas Locais	7	1.436,05	R\$ 2.927,05	0	6,550
16	Cluster + Tree + Buscas Locais	7	1.436,05	R\$ 2.927,05	0	6,560
17	GRASP - Vizinho mais Próximo - 0,10	6	1.653,19	R\$ 2.931,19	0	1029,490
18	Cluster + Nearest Neighbor + Buscas Locais	7	1.474,43	R\$ 2.965,43	0	7,790
19	Cluster + Boundary + Buscas Locais	7	1.475,42	R\$ 2.966,42	0	5,340
20	Nearest Pass + Buscas Locais	7	1.518,88	R\$ 3.009,88	0	3,710
21	GRASP - Vizinho mais Próximo - 0,15	6	1.745,00	R\$ 3.023,00	0	904,320
22	GRASP - Economias - versão 2 - 0,10	7	1.542,44	R\$ 3.033,44	0	977,670
23	Nearest Pass	7	1.549,50	R\$ 3.040,52	0	-
24	Nearest Neighbor + Buscas Locais	7	1.568,96	R\$ 3.059,96	0	5,600
25	Vizinho mais Próximo	7	1.573,41	R\$ 3.064,41	0	0,030
26	Economias - versão 2 + Buscas Locais	7	1.581,83	R\$ 3.072,83	0	18,370
27	GRASP - Economias - versão 2 - 0,20	7	1.627,62	R\$ 3.118,62	0	915,770
28	Cluster + Weighted Constraint	8	1.421,40	R\$ 3.125,40	0	-

29	GRASP - Vizinho mais Próximo - 0,20	7	1.678,20	R\$ 3.169,20	0	767,310
30	Cluster + Savings	8	1.607,90	R\$ 3.311,88	0	-
31	Cluster + Tree	8	1.607,90	R\$ 3.311,88	0	-
32	Economias - versão 1 + Buscas Locais	8	1.748,39	R\$ 3.452,39	0	6,640
33	Pie + Buscas Locais	8	1.543,05	R\$ 3.626,10	0	8,830
34	Cluster + Pie + Buscas Locais	7	1.876,43	R\$ 3.716,87	0	8,420
35	Cluster + Nearest Pass	11	1.530,00	R\$ 3.873,00	0	-
36	Nearest Neighbor	12	1.454,30	R\$ 4.010,26	0	-
37	Economias - versão 2	10	1.432,74	R\$ 4.014,66	0	12,020
38	Cluster + Boundary	12	1.530,00	R\$ 4.085,96	0	-
39	Economias - versão 1	11	1.744,51	R\$ 4.087,51	0	0,570
40	Cluster + Nearest Neighbor	13	1.656,12	R\$ 4.425,18	0	-
41	Pie	17	1.870,30	R\$ 7.838,74	0	-
42	Cluster + Pie	17	2.119,10	R\$ 7.961,36	0	-

Tabela 26: Resultados instância 155a

TABELAS DE RESULTADOS - instância 155a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,10	14	1.903,72	R\$ 5.244,82	0	1353,670
02	GRASP - Economias - versão 2 - 0,05	13	1.903,09	R\$ 5.267,52	0	1399,150
03	GRASP - Economias - versão 2 - 0,15	15	1.902,10	R\$ 5.450,86	0	1359,260
04	GRASP - Economias - versão 2 - 0,20	14	2.090,92	R\$ 5.468,88	0	1207,280
05	Cluster + Boundary + Buscas Locais	18	1.797,31	R\$ 5.730,21	0	8,620
06	Pie + Buscas Locais	13	1.933,76	R\$ 5.758,62	0	5,790
07	Cluster + Tree + Buscas Locais	17	2.036,18	R\$ 5.766,70	0	7,440
08	Economias - versão 2 + Buscas Locais	13	2.304,75	R\$ 5.772,47	0	21,330
09	Boundary	19	1.753,34	R\$ 5.800,39	0	-
10	Cluster + Nearest Neighbor + Buscas Locais	18	1.851,35	R\$ 5.812,40	0	9,580
11	Cluster + Nearest Pass + Buscas Locais	18	1.864,74	R\$ 5.829,18	0	9,180
12	Savings + Buscas Locais	19	1.790,30	R\$ 5.837,30	0	3,720
13	GRASP - Vizinho mais Próximo - 0,05	18	2.020,45	R\$ 5.854,45	0	1025,410
14	Weighted Constraint + Buscas Locais	19	1.863,12	R\$ 5.910,12	0	8,480
15	Economias - versão 1 + Buscas Locais	18	1.949,45	R\$ 5.914,17	0	7,680
16	Cluster + Pie + Buscas Locais	15	2.157,50	R\$ 5.940,80	0	7,150
17	GRASP - Vizinho mais Próximo - 0,10	18	2.180,96	R\$ 6.014,96	0	826,060
18	Weighted Constraint	19	1.974,90	R\$ 6.021,89	0	-
19	Tree + Buscas Locais	20	1.789,06	R\$ 6.049,06	0	4,620
20	Nearest Pass + Buscas Locais	19	2.008,66	R\$ 6.055,66	0	6,420
21	GRASP - Vizinho mais Próximo - 0,15	18	2.231,18	R\$ 6.065,18	0	753,910
22	Boundary + Buscas Locais	18	2.254,15	R\$ 6.088,15	0	4,400
23	GRASP - Vizinho mais Próximo - 0,20	17	2.358,97	R\$ 6.111,95	0	697,230
24	Savings	19	2.094,90	R\$ 6.141,87	0	-
25	Nearest Neighbor + Buscas Locais	19	2.024,21	R\$ 6.174,68	0	5,040
26	Economias - versão 2	17	1.682,93	R\$ 6.224,15	0	15,460
27	Tree	20	2.047,80	R\$ 6.307,79	0	-
28	Vizinho mais Próximo + Buscas Locais	18	2.504,58	R\$ 6.338,58	0	5,250

29	Nearest Pass	19	2.340,60	R\$ 6.387,60	0	-
30	Vizinho mais Próximo	19	2.673,12	R\$ 6.720,12	0	0,070
31	Economias - versão 1	20	2.090,14	R\$ 6.756,37	0	0,370
32	Cluster + Tree	21	2.266,20	R\$ 6.819,23	0	-
33	Pie	17	1.916,10	R\$ 7.422,13	0	-
34	Cluster + Pie	22	2.381,00	R\$ 8.720,73	0	-
35	Cluster + Savings + Buscas Locais	16	2.006,91	R\$ 5.519,02	1	8,560
36	Cluster + Weighted Constraint + Buscas Locais	18	1.681,68	R\$ 5.626,88	1	10,120
37	Cluster + Weighted Constraint	21	1.874,30	R\$ 6.427,29	1	-
38	Nearest Neighbor	21	2.008,30	R\$ 6.565,55	1	-
39	Cluster + Boundary	21	2.014,80	R\$ 6.582,74	1	-
40	Cluster + Nearest Neighbor	21	2.101,00	R\$ 6.673,10	1	-
41	Cluster + Savings	21	2.231,40	R\$ 6.784,49	1	-
42	Cluster + Nearest Pass	21	2.228,70	R\$ 6.804,07	1	-

Tabela 27: Resultados instância 155b

TABELAS DE RESULTADOS - instância 155b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,05	10	1.865,41	R\$ 4.462,24	0	1336,010
02	GRASP - Economias - versão 2 - 0,10	11	1.701,06	R\$ 4.538,13	0	1365,620
03	Economias - versão 2 + Buscas Locais	9	1.922,43	R\$ 4.608,46	0	21,130
04	GRASP - Economias - versão 2 - 0,15	11	1.650,65	R\$ 4.640,78	0	1316,320
05	Pie + Buscas Locais	10	1.881,96	R\$ 4.736,86	0	6,790
06	Cluster + Boundary + Buscas Locais	14	1.836,29	R\$ 4.818,29	0	9,820
07	Boundary + Buscas Locais	14	1.840,86	R\$ 4.822,86	0	8,390
08	Boundary	15	1.692,60	R\$ 4.887,63	0	-
09	GRASP - Economias - versão 2 - 0,20	11	1.829,48	R\$ 4.897,63	0	1417,270
10	Cluster + Tree + Buscas Locais	13	2.157,51	R\$ 4.926,51	0	8,210
11	Cluster + Savings + Buscas Locais	13	2.252,98	R\$ 5.021,98	0	5,120
12	GRASP - Vizinho mais Próximo - 0,05	14	2.052,25	R\$ 5.034,25	0	1363,090
13	GRASP - Vizinho mais Próximo - 0,10	14	2.073,08	R\$ 5.055,08	0	1076,120
14	Vizinho mais Próximo + Buscas Locais	14	2.089,56	R\$ 5.071,56	0	6,300
15	Cluster + Weighted Constraint	15	1.891,50	R\$ 5.086,54	0	-
16	Weighted Constraint + Buscas Locais	14	2.116,95	R\$ 5.098,95	0	8,910
17	Cluster + Nearest Pass + Buscas Locais	14	2.129,60	R\$ 5.111,60	0	6,540
18	Tree + Buscas Locais	16	1.748,42	R\$ 5.156,42	0	11,760
19	Cluster + Nearest Neighbor + Buscas Locais	15	1.996,39	R\$ 5.191,39	0	8,940
20	GRASP - Vizinho mais Próximo - 0,15	14	2.210,10	R\$ 5.192,10	0	934,200
21	Weighted Constraint	15	2.029,50	R\$ 5.224,49	0	-
22	Savings + Buscas Locais	16	1.827,55	R\$ 5.235,55	0	8,610
23	Cluster + Pie + Buscas Locais	12	1.951,70	R\$ 5.243,43	0	8,300
24	Cluster + Savings	15	2.058,20	R\$ 5.253,21	0	-
25	Economias - versão 2	13	1.620,17	R\$ 5.255,59	0	16,850
26	GRASP - Vizinho mais Próximo - 0,20	14	2.327,80	R\$ 5.309,80	0	837,110
27	Nearest Pass + Buscas Locais	16	1.912,42	R\$ 5.320,42	0	12,060
28	Cluster + Weighted Constraint + Buscas Locais	14	2.386,92	R\$ 5.368,92	0	7,730

29	Tree	16	1.995,30	R\$ 5.403,31	0	-
30	Savings	16	2.071,20	R\$ 5.479,21	0	-
31	Nearest Neighbor + Buscas Locais	17	1.863,90	R\$ 5.484,90	0	13,830
32	Cluster + Boundary	17	1.892,50	R\$ 5.513,53	0	-
33	Cluster + Tree	16	2.118,80	R\$ 5.526,82	0	-
34	Economias - versão 1 + Buscas Locais	13	2.271,38	R\$ 5.581,80	0	5,340
35	Nearest Pass	16	2.281,60	R\$ 5.689,58	0	-
36	Cluster + Nearest Pass	17	2.239,40	R\$ 5.860,40	0	-
37	Vizinho mais Próximo	15	2.698,35	R\$ 5.893,35	0	0,080
38	Economias - versão 1	17	2.243,00	R\$ 6.339,04	0	0,410
39	Cluster + Nearest Neighbor	21	2.124,00	R\$ 6.668,98	0	-
40	Nearest Neighbor	21	2.158,90	R\$ 6.703,87	0	-
41	Pie	16	1.888,10	R\$ 7.052,45	0	-
42	Cluster + Pie	20	2.167,10	R\$ 8.392,03	0	-

Tabela 28: Resultados instância 170a

TABELAS DE RESULTADOS - instância 170a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Economias - versão 2 + Buscas Locais	14	1.714,07	R\$ 5.606,32	0	36,750
02	GRASP - Economias - versão 2 - 0,05	14	1.709,92	R\$ 5.613,48	0	1822,737
03	GRASP - Economias - versão 2 - 0,10	15	2.232,19	R\$ 5.935,50	0	1935,797
04	Pie + Buscas Locais	14	1.905,07	R\$ 6.056,73	0	8,830
05	Boundary + Buscas Locais	18	1.950,13	R\$ 6.071,42	0	5,180
06	Savings + Buscas Locais	18	2.098,23	R\$ 6.190,56	0	4,290
07	GRASP - Economias - versão 2 - 0,15	15	2.265,90	R\$ 6.197,67	0	1940,707
08	Boundary	20	1.784,90	R\$ 6.218,91	0	-
09	Tree + Buscas Locais	19	2.056,34	R\$ 6.286,58	0	7,830
10	Cluster + Savings + Buscas Locais	18	2.279,88	R\$ 6.297,41	0	4,860
11	Economias - versão 2	17	1.716,92	R\$ 6.337,43	0	26,500
12	Cluster + Weighted Constraint + Buscas Locais	19	2.077,15	R\$ 6.340,10	0	6,330
13	GRASP - Economias - versão 2 - 0,20	18	2.262,00	R\$ 6.422,31	0	1900,417
14	GRASP - Vizinho mais Próximo - 0,05	18	2.396,94	R\$ 6.461,19	0	1219,960
15	Cluster + Tree + Buscas Locais	18	2.398,60	R\$ 6.471,09	0	5,850
16	Cluster + Pie + Buscas Locais	17	2.132,86	R\$ 6.480,91	0	5,600
17	Savings	20	2.108,10	R\$ 6.543,30	0	-
18	Weighted Constraint + Buscas Locais	18	2.490,01	R\$ 6.555,70	0	6,050
19	Cluster + Boundary + Buscas Locais	20	2.149,03	R\$ 6.588,34	0	8,400
20	GRASP - Vizinho mais Próximo - 0,10	18	2.560,30	R\$ 6.659,22	0	1040,710
21	Cluster + Nearest Neighbor + Buscas Locais	18	2.324,41	R\$ 6.724,08	0	6,610
22	Nearest Pass + Buscas Locais	20	2.287,31	R\$ 6.737,82	0	8,550
23	GRASP - Vizinho mais Próximo - 0,15	18	2.684,82	R\$ 6.737,89	0	956,690
24	Weighted Constraint	19	2.545,50	R\$ 6.756,50	0	-
25	Economias - versão 1 + Buscas Locais	15	2.823,60	R\$ 6.783,21	0	4,570
26	Tree	21	2.173,10	R\$ 6.806,97	0	-
27	Cluster + Weighted Constraint	22	1.998,80	R\$ 6.845,67	0	-
28	GRASP - Vizinho mais Próximo - 0,20	18	2.762,30	R\$ 6.866,93	0	851,340

29	Cluster + Savings	21	2.261,70	R\$ 6.895,64	0	-
30	Cluster + Tree	21	2.269,10	R\$ 6.902,99	0	-
31	Cluster + Nearest Pass + Buscas Locais	18	2.815,99	R\$ 6.933,89	0	5,960
32	Vizinho mais Próximo + Buscas Locais	18	2.932,13	R\$ 7.185,57	0	8,160
33	Nearest Pass	22	2.707,40	R\$ 7.550,83	0	-
34	Vizinho mais Próximo	20	3.289,89	R\$ 7.707,28	0	0,110
35	Economias - versão 1	21	2.614,34	R\$ 7.878,68	0	0,700
36	Cluster + Pie	22	2.298,70	R\$ 8.073,70	0	-
37	Pie	19	2.036,00	R\$ 8.347,94	0	-
38	Nearest Neighbor + Buscas Locais	20	2.339,05	R\$ 6.800,74	1	11,400
39	Nearest Neighbor	23	2.122,30	R\$ 7.247,20	1	-
40	Cluster + Nearest Pass	23	2.768,00	R\$ 7.907,86	1	-
41	Cluster + Boundary	22	2.161,00	R\$ 7.025,80	2	-
42	Cluster + Nearest Neighbor	26	2.351,40	R\$ 8.346,80	8	-

Tabela 29: Resultados instância 170b

TABELAS DE RESULTADOS - instância 170b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,05	15	1.726,10	R\$ 6.244,85	0	1768,047
02	GRASP - Economias - versão 2 - 0,10	15	1.991,83	R\$ 6.601,08	0	1764,227
03	GRASP - Economias - versão 2 - 0,20	16	2.220,93	R\$ 6.826,44	0	1684,057
04	GRASP - Economias - versão 2 - 0,15	17	1.643,33	R\$ 6.829,84	0	1643,297
05	Economias - versão 2 + Buscas Locais	17	1.953,41	R\$ 6.842,81	0	41,780
06	Economias - versão 2	18	1.911,65	R\$ 6.912,99	0	28,280
07	Pie + Buscas Locais	15	2.338,99	R\$ 7.025,16	0	7,520
08	Savings + Buscas Locais	21	2.314,56	R\$ 7.072,35	0	8,450
09	Cluster + Weighted Constraint + Buscas Locais	20	2.548,16	R\$ 7.097,52	0	6,830
10	Cluster + Weighted Constraint	22	2.122,60	R\$ 7.121,29	0	-
11	GRASP - Vizinho mais Próximo - 0,10	20	2.538,84	R\$ 7.140,83	0	1155,700
12	GRASP - Vizinho mais Próximo - 0,05	20	2.562,55	R\$ 7.144,02	0	1360,890
13	Economias - versão 1 + Buscas Locais	17	2.638,33	R\$ 7.179,22	0	6,480
14	Cluster + Boundary + Buscas Locais	19	2.462,70	R\$ 7.191,90	0	8,610
15	Weighted Constraint + Buscas Locais	21	2.474,29	R\$ 7.240,30	0	9,390
16	Vizinho mais Próximo + Buscas Locais	20	2.590,75	R\$ 7.242,00	0	6,580
17	Weighted Constraint	21	2.560,80	R\$ 7.326,79	0	-
18	GRASP - Vizinho mais Próximo - 0,15	20	2.719,17	R\$ 7.327,44	0	1051,340
19	GRASP - Vizinho mais Próximo - 0,20	20	2.807,66	R\$ 7.409,65	0	950,380
20	Cluster + Nearest Pass + Buscas Locais	20	2.644,07	R\$ 7.410,76	0	6,370
21	Savings	22	2.412,40	R\$ 7.412,00	0	-
22	Cluster + Pie + Buscas Locais	19	2.415,13	R\$ 7.612,86	0	7,020
23	Nearest Pass + Buscas Locais	22	2.691,76	R\$ 7.663,37	0	10,100
24	Vizinho mais Próximo	21	3.024,91	R\$ 7.782,39	0	0,110
25	Nearest Pass	22	3.125,60	R\$ 8.096,08	0	-
26	Economias - versão 1	24	2.416,02	R\$ 8.240,11	0	0,980
27	Pie	21	2.243,80	R\$ 9.557,79	0	-
28	Cluster + Pie	25	2.708,10	R\$ 9.842,50	0	-

29	Cluster + Tree + Buscas Locais	19	2.503,68	R\$ 6.869,14	1	5,950
30	Cluster + Nearest Neighbor + Buscas Locais	19	2.312,86	R\$ 6.888,00	1	8,300
31	Nearest Neighbor + Buscas Locais	21	2.165,70	R\$ 6.923,51	1	7,090
32	Boundary	22	1.958,80	R\$ 6.929,30	1	-
33	Boundary + Buscas Locais	21	2.239,05	R\$ 6.996,63	1	9,750
34	Tree + Buscas Locais	22	2.079,85	R\$ 7.050,33	1	5,850
35	Cluster + Savings + Buscas Locais	19	2.763,48	R\$ 7.099,97	1	5,900
36	Tree	22	2.295,80	R\$ 7.291,53	1	-
37	Cluster + Tree	22	2.342,30	R\$ 7.340,93	1	-
38	Cluster + Savings	22	2.353,70	R\$ 7.352,33	1	-
39	Cluster + Nearest Pass	23	2.898,20	R\$ 8.277,91	2	-
40	Nearest Neighbor	23	2.274,90	R\$ 7.521,65	3	-
41	Cluster + Boundary	23	2.393,70	R\$ 7.691,99	4	-
42	Cluster + Nearest Neighbor	26	2.492,60	R\$ 8.657,41	15	-

Tabela 30: Resultados instância 185a

TABELAS DE RESULTADOS - instância 185a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,05	18	2.221,06	R\$ 7.685,95	0	1739,803
02	Economias - versão 2 + Buscas Locais	19	2.190,15	R\$ 7.771,92	0	88,190
03	Boundary + Buscas Locais	23	2.376,22	R\$ 7.841,06	0	9,730
04	GRASP - Economias - versão 2 - 0,10	19	2.624,03	R\$ 7.900,46	0	1688,643
05	Boundary	24	2.334,80	R\$ 7.977,00	0	-
06	Pie + Buscas Locais	19	2.329,43	R\$ 8.059,76	0	10,710
07	Savings + Buscas Locais	23	2.638,73	R\$ 8.071,69	0	7,570
08	GRASP - Economias - versão 2 - 0,15	21	2.512,48	R\$ 8.146,03	0	2135,053
09	GRASP - Economias - versão 2 - 0,20	19	2.661,88	R\$ 8.307,33	0	1985,413
10	Economias - versão 2	21	2.226,09	R\$ 8.352,36	0	75,550
11	Cluster + Nearest Neighbor + Buscas Locais	22	2.727,53	R\$ 8.438,87	0	9,440
12	Savings	25	2.692,60	R\$ 8.552,79	0	-
13	Nearest Pass + Buscas Locais	23	3.089,66	R\$ 8.625,20	0	10,220
14	Cluster + Boundary + Buscas Locais	21	2.879,18	R\$ 8.635,70	0	7,920
15	Weighted Constraint	24	3.119,00	R\$ 8.768,27	0	-
16	Economias - versão 1 + Buscas Locais	21	2.853,30	R\$ 8.801,22	0	9,270
17	Cluster + Nearest Pass + Buscas Locais	21	2.974,57	R\$ 8.850,84	0	7,730
18	Cluster + Pie + Buscas Locais	22	2.978,19	R\$ 9.184,78	0	5,130
19	Nearest Pass	25	3.310,90	R\$ 9.197,19	0	-
20	GRASP - Vizinho mais Próximo - 0,20	23	3.253,65	R\$ 9.291,64	0	1063,320
21	GRASP - Vizinho mais Próximo - 0,15	23	3.397,09	R\$ 9.321,55	0	1128,670
22	GRASP - Vizinho mais Próximo - 0,05	23	3.238,46	R\$ 9.339,44	0	1529,500
23	Weighted Constraint + Buscas Locais	22	3.716,58	R\$ 9.533,28	0	4,910
24	GRASP - Vizinho mais Próximo - 0,10	23	3.507,46	R\$ 9.581,29	0	1287,820
25	Economias - versão 1	24	3.030,48	R\$ 9.952,44	0	1,100
26	Vizinho mais Próximo + Buscas Locais	24	3.718,79	R\$ 10.007,96	0	8,270
27	Pie	23	2.470,96	R\$ 10.291,96	0	-
28	Vizinho mais Próximo	25	3.939,96	R\$ 10.337,98	0	0,130

29	Cluster + Savings + Buscas Locais	21	2.852,36	R\$ 8.000,47	1	7,060
30	Cluster + Tree + Buscas Locais	21	2.852,36	R\$ 8.000,47	1	7,060
31	Tree + Buscas Locais	24	2.384,14	R\$ 8.017,89	1	10,520
32	Cluster + Savings	25	2.568,90	R\$ 8.407,49	1	-
33	Cluster + Tree	25	2.568,90	R\$ 8.407,49	1	-
34	Tree	25	2.634,00	R\$ 8.503,51	1	-
35	Cluster + Weighted Constraint	25	2.688,10	R\$ 8.527,86	1	-
36	Cluster + Weighted Constraint + Buscas Locais	21	3.194,83	R\$ 9.016,77	1	6,830
37	Nearest Neighbor + Buscas Locais	24	2.466,15	R\$ 8.127,20	3	13,520
38	Nearest Neighbor	26	2.602,80	R\$ 8.725,32	3	-
39	Cluster + Nearest Pass	26	2.883,50	R\$ 9.706,39	14	-
40	Cluster + Boundary	27	2.739,50	R\$ 9.728,84	14	-
41	Cluster + Nearest Neighbor	30	2.793,70	R\$ 10.642,03	14	-
42	Cluster + Pie	28	2.970,10	R\$ 10.901,31	14	-

Tabela 31: Resultados instância 185b

TABELAS DE RESULTADOS - instância 185b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Economias - versão 2 + Buscas Locais	13	1.395,05	R\$ 4.700,94	0	58,730
02	GRASP - Economias - versão 2 - 0,05	12	1.510,74	R\$ 4.721,77	0	196,733
03	GRASP - Economias - versão 2 - 0,10	14	1.508,96	R\$ 4.957,73	0	98,463
04	Pie + Buscas Locais	12	1.637,92	R\$ 5.016,97	0	6,950
05	GRASP - Economias - versão 2 - 0,15	13	1.857,93	R\$ 5.017,73	0	169,513
06	GRASP - Economias - versão 2 - 0,20	14	1.811,68	R\$ 5.082,48	0	608,003
07	Cluster + Pie + Buscas Locais	15	1.718,70	R\$ 5.268,70	0	5,070
08	Cluster + Nearest Neighbor + Buscas Locais	17	1.601,83	R\$ 5.317,53	0	5,030
09	Cluster + Nearest Pass + Buscas Locais	17	1.599,20	R\$ 5.337,80	0	9,600
10	Economias - versão 2	16	1.375,15	R\$ 5.363,71	0	48,500
11	Cluster + Tree + Buscas Locais	17	1.667,16	R\$ 5.403,06	0	6,910
12	GRASP - Vizinho mais Próximo - 0,05	17	1.802,01	R\$ 5.423,01	0	1934,360
13	Cluster + Savings + Buscas Locais	17	1.687,82	R\$ 5.424,46	0	5,630
14	Economias - versão 1 + Buscas Locais	17	1.690,92	R\$ 5.427,29	0	8,130
15	Cluster + Boundary + Buscas Locais	16	1.786,75	R\$ 5.452,94	0	2,740
16	Boundary + Buscas Locais	20	1.305,79	R\$ 5.565,79	0	12,960
17	Boundary	20	1.334,30	R\$ 5.594,32	0	-
18	GRASP - Vizinho mais Próximo - 0,15	17	1.973,79	R\$ 5.594,79	0	1496,170
19	Vizinho mais Próximo + Buscas Locais	18	1.777,03	R\$ 5.611,03	0	10,150
20	GRASP - Vizinho mais Próximo - 0,10	18	1.780,57	R\$ 5.614,57	0	1554,500
21	Cluster + Weighted Constraint + Buscas Locais	20	1.449,26	R\$ 5.709,26	0	7,340
22	Weighted Constraint + Buscas Locais	18	1.947,80	R\$ 5.781,80	0	4,330
23	GRASP - Vizinho mais Próximo - 0,20	18	1.848,44	R\$ 5.799,82	0	1307,230
24	Cluster + Weighted Constraint	20	1.569,50	R\$ 5.829,45	0	-
25	Savings + Buscas Locais	20	1.582,86	R\$ 5.842,86	0	9,470
26	Weighted Constraint	19	1.801,20	R\$ 5.848,20	0	-
27	Nearest Pass + Buscas Locais	20	1.608,91	R\$ 5.868,91	0	14,310
28	Savings	20	1.768,20	R\$ 6.028,24	0	-

29	Vizinho mais Próximo	19	2.041,60	R\$ 6.088,60	0	0,100
30	Nearest Pass	20	1.830,40	R\$ 6.090,40	0	-
31	Cluster + Boundary	21	1.711,60	R\$ 6.260,33	0	-
32	Cluster + Tree	21	1.770,50	R\$ 6.316,74	0	-
33	Cluster + Savings	21	1.771,90	R\$ 6.318,12	0	-
34	Cluster + Nearest Pass	21	1.765,10	R\$ 6.318,63	0	-
35	Economias - versão 1	21	1.732,01	R\$ 6.446,52	0	0,770
36	Cluster + Pie	21	1.798,00	R\$ 7.047,94	0	-
37	Pie	18	1.577,20	R\$ 7.129,84	0	-
38	Nearest Neighbor + Buscas Locais	18	1.429,54	R\$ 5.394,37	1	15,180
39	Tree + Buscas Locais	20	1.409,19	R\$ 5.669,19	1	7,490
40	Tree	20	1.564,80	R\$ 5.824,76	1	-
41	Nearest Neighbor	21	1.584,60	R\$ 6.133,54	2	-
42	Cluster + Nearest Neighbor	24	1.721,50	R\$ 7.133,88	2	-

Tabela 32: Resultados instância 200a

TABELAS DE RESULTADOS - instância 200a						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	GRASP - Economias - versão 2 - 0,05	18	2.608,82	R\$ 7.129,91	0	2131,685
02	GRASP - Economias - versão 2 - 0,10	18	2.275,96	R\$ 7.278,01	0	1837,362
03	GRASP - Economias - versão 2 - 0,15	20	2.770,35	R\$ 7.598,03	0	2354,554
04	GRASP - Economias - versão 2 - 0,20	21	2.893,99	R\$ 7.632,98	0	2500,647
05	Pie + Buscas Locais	19	2.480,14	R\$ 7.655,87	0	13,500
06	GRASP - Vizinho mais Próximo - 0,05	21	2.756,84	R\$ 7.924,25	0	2018,560
07	Economias - versão 2 + Buscas Locais	22	2.363,38	R\$ 7.969,19	0	167,640
08	Cluster + Pie + Buscas Locais	21	2.803,82	R\$ 8.137,57	0	15,690
09	GRASP - Vizinho mais Próximo - 0,15	22	2.902,97	R\$ 8.233,22	0	1373,590
10	GRASP - Vizinho mais Próximo - 0,20	21	3.324,36	R\$ 8.288,87	0	1299,700
11	GRASP - Vizinho mais Próximo - 0,10	22	2.938,73	R\$ 8.305,91	0	1601,370
12	Economias - versão 1 + Buscas Locais	20	3.144,75	R\$ 8.351,61	0	8,810
13	Vizinho mais Próximo + Buscas Locais	22	3.069,34	R\$ 8.464,13	0	11,810
14	Economias - versão 2	24	2.424,43	R\$ 8.508,28	0	155,480
15	Vizinho mais Próximo	23	3.351,10	R\$ 9.160,29	0	0,140
16	Pie	24	2.315,90	R\$ 10.077,73	0	-
17	Cluster + Nearest Pass + Buscas Locais	21	2.881,89	R\$ 7.486,83	1	15,790
18	Nearest Pass + Buscas Locais	20	3.150,55	R\$ 7.562,22	1	10,830
19	Nearest Pass	22	3.471,30	R\$ 8.314,01	1	-
20	Cluster + Pie	27	2.748,50	R\$ 10.417,02	2	-
21	Nearest Neighbor + Buscas Locais	18	2.526,48	R\$ 6.597,41	3	8,010
22	Cluster + Boundary + Buscas Locais	18	2.773,02	R\$ 6.791,86	3	7,990
23	Cluster + Nearest Neighbor + Buscas Locais	20	2.584,50	R\$ 6.966,92	3	16,430
24	Savings + Buscas Locais	18	2.218,20	R\$ 6.188,84	5	10,220
25	Cluster + Savings + Buscas Locais	17	2.416,47	R\$ 6.198,02	5	6,740
26	Cluster + Weighted Constraint + Buscas Locais	18	2.260,84	R\$ 6.229,74	5	9,320
27	Cluster + Tree + Buscas Locais	17	2.486,05	R\$ 6.267,61	5	9,130
28	Cluster + Weighted Constraint	21	2.189,00	R\$ 6.736,23	5	-

29	Weighted Constraint + Buscas Locais	20	2.560,34	R\$ 6.820,34	5	10,890
30	Savings	21	2.290,15	R\$ 6.837,74	5	-
31	Weighted Constraint	20	2.625,00	R\$ 6.885,04	5	-
32	Cluster + Savings	21	2.508,90	R\$ 7.060,05	5	-
33	Cluster + Tree	21	2.533,10	R\$ 7.084,24	5	-
34	Tree + Buscas Locais	16	2.242,48	R\$ 5.803,37	6	6,830
35	Boundary + Buscas Locais	18	2.229,60	R\$ 6.224,87	6	14,170
36	Boundary	21	2.046,30	R\$ 6.605,27	6	-
37	Tree	21	2.324,30	R\$ 6.871,52	6	-
38	Nearest Neighbor	24	2.381,20	R\$ 7.811,77	6	-
39	Cluster + Boundary	21	2.426,40	R\$ 6.988,77	7	-
40	Cluster + Nearest Pass	22	3.122,90	R\$ 8.142,49	9	-
41	Cluster + Nearest Neighbor	25	2.557,60	R\$ 8.293,82	12	-
42	Economias - versão 1	24	2.829,91	R\$ 8.990,00	12	1,330

Tabela 33: Resultados instância 200b

TABELAS DE RESULTADOS - instância 200b						
Rank	Métodos	Rotas	Kilometros	Custo	Não Roteados	Tempo (seg.)
01	Economias - versão 2 + Buscas Locais	17	1.576,22	R\$ 5.951,43	0	117,960
02	GRASP - Economias - versão 2 - 0,05	17	1.576,22	R\$ 5.951,43	0	2517,960
03	Cluster + Nearest Neighbor + Buscas Locais	19	1.843,45	R\$ 5.995,81	0	13,170
04	GRASP - Economias - versão 2 - 0,15	17	1.585,47	R\$ 5.998,90	0	2641,680
05	GRASP - Economias - versão 2 - 0,10	19	1.641,65	R\$ 6.014,95	0	2359,500
06	Pie + Buscas Locais	16	2.172,73	R\$ 6.372,31	0	11,300
07	Cluster + Pie + Buscas Locais	17	2.229,80	R\$ 6.376,56	0	9,290
08	GRASP - Economias - versão 2 - 0,20	21	1.832,32	R\$ 6.463,64	0	2865,892
09	GRASP - Vizinho mais Próximo - 0,05	19	2.253,22	R\$ 6.520,88	0	2083,557
10	GRASP - Vizinho mais Próximo - 0,10	19	2.273,65	R\$ 6.543,39	0	1854,520
11	Economias - versão 2	20	1.597,89	R\$ 6.602,89	0	100,340
12	GRASP - Vizinho mais Próximo - 0,15	19	2.362,82	R\$ 6.680,74	0	1581,730
13	GRASP - Vizinho mais Próximo - 0,20	20	2.303,30	R\$ 6.748,81	0	1430,260
14	Vizinho mais Próximo + Buscas Locais	21	2.320,67	R\$ 7.076,53	0	12,120
15	Cluster + Pie	23	2.080,40	R\$ 7.596,03	0	-
16	Vizinho mais Próximo	22	2.624,40	R\$ 7.674,73	0	0,150
17	Pie	23	1.834,80	R\$ 8.406,07	0	-
18	Cluster + Nearest Pass + Buscas Locais	18	1.970,02	R\$ 5.908,97	1	9,340
19	Savings + Buscas Locais	19	1.775,61	R\$ 5.939,56	1	6,140
20	Tree + Buscas Locais	20	1.649,11	R\$ 6.017,14	1	13,460
21	Cluster + Boundary + Buscas Locais	18	2.183,95	R\$ 6.177,44	1	5,130
22	Weighted Constraint + Buscas Locais	19	2.193,29	R\$ 6.361,49	1	12,220
23	Tree	21	1.828,40	R\$ 6.390,75	1	-
24	Economias - versão 1 + Buscas Locais	18	2.311,60	R\$ 6.454,96	1	10,580
25	Boundary + Buscas Locais	18	1.663,13	R\$ 5.646,75	2	9,660
26	Cluster + Weighted Constraint + Buscas Locais	18	1.735,02	R\$ 5.689,37	2	7,840
27	Nearest Neighbor + Buscas Locais	19	1.655,40	R\$ 5.804,41	2	11,910
28	Boundary	21	1.581,40	R\$ 6.131,97	2	-

29	Cluster + Weighted Constraint	21	1.778,70	R\$ 6.329,25	2	-
30	Savings	22	1.900,70	R\$ 6.730,97	2	-
31	Weighted Constraint	21	2.270,10	R\$ 6.809,02	2	-
32	Nearest Pass + Buscas Locais	18	1.875,67	R\$ 5.875,76	3	6,980
33	Cluster + Savings + Buscas Locais	18	1.963,49	R\$ 5.958,28	3	13,460
34	Cluster + Tree + Buscas Locais	18	1.963,49	R\$ 5.958,28	3	13,450
35	Cluster + Nearest Pass	21	1.958,90	R\$ 6.526,87	3	-
36	Cluster + Boundary	21	1.963,30	R\$ 6.527,75	3	-
37	Cluster + Savings	21	2.007,60	R\$ 6.569,99	3	-
38	Cluster + Tree	21	2.007,60	R\$ 6.569,99	3	-
39	Nearest Neighbor	22	1.760,80	R\$ 6.596,43	3	-
40	Nearest Pass	21	2.097,20	R\$ 6.665,13	3	-
41	Cluster + Nearest Neighbor	24	1.928,20	R\$ 7.337,47	3	-
42	Economias - versão 1	22	1.790,92	R\$ 6.714,27	17	2,520