

DÂNGELO SILVA MIRANDA

**APLICAÇÃO DE METAHEURÍSTICAS PARA O PROBLEMA DE ROTEAMENTO
DE VEÍCULOS DINÂMICO PARA TRANSPORTE REATIVO A DEMANDA**

**Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título de
Magister Scientiae.**

**VIÇOSA
MINAS GERAIS - BRASIL
2012**

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

M672a
2012

Miranda, Dângelo Silva, 1984-

Aplicação de metaheurísticas para o problema de roteamento
de veículos dinâmico para transporte reativo a demanda /
Dângelo Silva Miranda. – Viçosa, MG, 2012.
xiii, 77f. : il. ; (algumas color.) ; 29cm.

Inclui apêndices.

Orientador: André Gustavo dos Santos

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 65-70

1. Pesquisa operacional. 2. Otimização combinatória.
3. Otimização matemática. 4. Programação heurística.
5. Logística. I. Universidade Federal de Viçosa. Departamento
de Informática. Programa de Pós-Graduação em Ciência da
Computação. II. Título.

CDD 22. ed. 003

DÂNGELO SILVA MIRANDA

**APLICAÇÃO DE METAHEURÍSTICAS PARA O PROBLEMA DE
ROTEAMENTO DE VEÍCULOS DINÂMICO PARA TRANSPORTE
REATIVO A DEMANDA**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 09 de agosto de 2012.

Profº José Elias Claudio Arroyo
(Coorientador)

Profº Luiz Aurélio Raggi

Profº André Gustavo dos Santos
(Orientador)

*Dedico essa dissertação ao meu grande Deus,
O Todo-Poderoso*

*Aos meus pais
João Bosco, Irenice*

*À minha esposa
Gislane*

E à toda minha família e amigos

AGRADECIMENTOS

Graças lhe dou SENHOR não apenas por mais esta conquista, mas porque o SENHOR se faz presente na minha vida e de minha família sendo a ROCHA de nossas vidas e em todo tempo nos garante o refrigério nos tempos de angústia, vitória nos tempos de batalha e a firmeza de que o SENHOR cumpre todas as promessas feitas aos seus filhos. Todas as coisas só foram possíveis mediante a fê outorgada pelo SENHOR, o Deus que vivifica mortos e chama a existência coisas que não existem. A Ti SENHOR, seja toda honra, glória e louvor. Amém.

Aos meus pais João Bosco de Miranda e Irenice Maria Silva Miranda pelo amor incondicional que sempre me deram, a motivação e incentivo a persistir nos objetivos, através de palavras restauradoras.

À minha amada esposa Gislane Natália de Souza Miranda que soube iluminar a minha vida, sendo minha amiga, companheira e amante inseparável. Obrigado por tudo meu anjo.

Ao meu orientador André Gustavo dos Santos que me recebeu como orientado, empenhou-se pela realização desse trabalho e apoiou a exploração desse assunto. Sempre me motivou a persistir seja através dos momentos de descontração ou uma palavra mais severa. Obrigado pela paciência, mostrando a direção nos momentos em que eu estava um pouco “perdido” durante o andamento e desenvolvimento das atividades.

A todos os professores e funcionários pela dedicação e carinho transmitidos durante todo o percurso.

Aos meus amigos e principalmente a Igreja em Viçosa, por me acolherem desde a trajetória da graduação.

À Reitoria, Pró-Reitoria de Assuntos Comunitários, Conselho Comunitário e todos que estiveram envolvidos na criação das condições que asseguraram minha permanência no mestrado.

À Universidade Federal de Viçosa que me acolhe desde a graduação e disponibilizou sua estrutura para realização deste trabalho.

BIOGRAFIA

DÂNGELO SILVA MIRANDA, nascido no dia 13 de Junho do ano de 1984, natural da cidade de Ipatinga no estado de Minas Gerais, é filho de João Bosco de Miranda e Irenice Maria Silva Miranda.

Terminou o ensino médio no Colégio Tiradentes da Polícia Militar de Ipatinga/MG em 2002/2 e ingressou na Universidade Federal de Viçosa em 2004/1 para fazer o curso de Bacharel em Ciência da Computação.

Trabalhou com serviços técnicos em computação como estagiário e pôde colocar em prática alguns dos conhecimentos adquiridos durante a graduação.

Após conclusão do bacharelado em 2009/1, foi selecionado para ingressar no Mestrado em Ciência da Computação da UFV no segundo semestre do mesmo ano.

Trabalhou como professor convidado na faculdade União do Ensino Superior de Viçosa (UNIVIÇOSA) em Viçosa/MG durante o segundo semestre de 2009.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE TABELAS	ix
LISTA DE ABREVIATURAS E SIGLAS.....	xi
RESUMO	xii
ABSTRACT	xiii
1 INTRODUÇÃO	1
1.1 Motivação	2
1.2 Objetivos	4
1.3 Organização do documento	5
2 REFERENCIAL TEÓRICO.....	6
2.1 Classificação do problema.....	7
2.2 Transporte Reativo a Demanda	7
2.3 Trabalhos Relacionados.....	11
3 DEFINIÇÃO DO PROBLEMA.....	14
3.1 Definição do Problema	14
4 MÉTODOS HEURÍSTICOS PROPOSTOS	20
4.1 Representação de uma solução do problema.....	21
4.2 Metaheurística GRASP	22
4.2.1 Construção de soluções viáveis.....	24
4.2.2 Método GRASP Reativo	27
4.3 Busca Local	29
4.4 GRASP Reativo + Busca Tabu	33
4.5 Replanejamento	36
5 EXPERIMENTAÇÃO COMPUTACIONAL	41
5.1 Ambiente de testes e instâncias utilizadas.....	41
5.2 Parâmetros dos métodos propostos e sua utilização nos testes	42
5.2.1 Parâmetros das funções de avaliação	42
5.2.2 Parâmetros do GRASP Reativo	43
5.2.3 Parâmetros do GRASP Reativo + Busca Tabu	43
5.2.4 Parâmetros dos Replanejamentos.....	43
5.3 Análise e discussão dos resultados dos experimentos computacionais.....	44

5.3.1	Tipos de Estratégia e Movimento	44
5.3.2	Quantidade e Capacidade dos Veículos	47
5.3.3	Métodos para o caso estático.....	50
5.3.4	Métodos para o caso dinâmico	54
6	CONCLUSÕES	62
6.1	Trabalhos Futuros	63
	REFERÊNCIAS BIBLIOGRÁFICAS	65
	APÊNDICE A.....	71
	APÊNDICE B.....	75

LISTA DE FIGURAS

Figura 1 - Exemplo do critério Flexibilidade. Fonte: (JOSSELIN <i>et al.</i> , 2008).	8
Figura 2 - Exemplo do critério Qualidade de Serviço. Fonte: (JOSSELIN <i>et al.</i> , 2008).	9
Figura 3 - Exemplo do critério Eficiência. Fonte: (JOSSELIN <i>et al.</i> , 2008).	9
Figura 4 - Estimativa do custo médio por viagem em função das viagens por dia. Fonte: (JOKINEN <i>et al.</i> , 2011).	10
Figura 5 - Sistema TRD é "resilient". Fonte: (JOKINEN <i>et al.</i> , 2011).	11
Figura 6 - Estado inicial hipotético para um PRVDTRD estático. Adaptado de (GOMES <i>et al.</i> , 2010).	17
Figura 7 - Representação da solução para problema proposto na Figura 6.	19
Figura 8 - Algoritmo GRASP. Adaptado de (FEO; RESENDE, 1995).	24
Figura 9 - Algoritmo da fase de construção do GRASP Reativo. Adaptado de (GOMES <i>et al.</i> , 2010).	26
Figura 10 - Algoritmo GRASP_Reativo. Adaptado de (KAMPKE, 2010).	28
Figura 11 - Representação de uma solução com duas rotas.	29
Figura 12 - Representação concatenada de uma solução com duas rotas.	29
Figura 13 - Exemplo da aplicação do movimento Troca.	30
Figura 14 - Exemplo da aplicação do movimento Substituição.	30
Figura 15 - Exemplo da aplicação do movimento Deslocamento à esquerda.	30
Figura 16 - Exemplo da aplicação do movimento Deslocamento à direita.	30
Figura 17 - Busca Local com tipos de estratégias diferentes.	32
Figura 18 - Estratégia Melhor Vizinho com tipo de movimento genérico.	33
Figura 19 - Estratégia Primeiro Melhor Vizinho com tipo de movimento genérico.	33
Figura 20 - Algoritmo da Busca Tabu.	35
Figura 21 - Exemplo de rota concatenada semelhante ao apresentado na Figura 12.	35
Figura 22 - Exemplo da aplicação do movimento Substituição dentro da Busca Tabu.	36
Figura 23 - Exemplo de Lista Tabu iteração 1.	36
Figura 24 - Algoritmo do replanejamento RepGuloso.	38
Figura 25 - Algoritmo do replanejamento RepGR.	40
Figura 26 - Comparação entre as estratégias Primeiro Melhor e Melhor Vizinho.	46
Figura 27 - Comparação entre os movimentos Troca, Substituição, Deslocamento e Aleatório.	47
Figura 28 - Influência da quantidade de veículos sobre FO Médio.	48
Figura 29 - Influência da capacidade do veículo sobre FO Médio.	49
Figura 30 - Média dos valores de FO Médio considerando apenas Frota e Capacidade.	50

Figura 31 - Comparação entre as heurísticas Construtiva Gulosa (GOMES <i>et al.</i> , 2010), GRASP-like (GOMES <i>et al.</i> , 2010) e GRASP Reativo proposto.....	52
Figura 32 - Comparação entre GRASP Reativo + BT, GRASP Reativo e Melhor Resultado Encontrado de cada instância para o caso estático sob os critérios de Melhor FO e FO Médio (não considerando GRASP Reativo + BT).	53
Figura 33 - Replanejamento com solução inicial da heurística GRASP Reativo + Busca Tabu.	56
Figura 34 - Comparação entre as heurísticas utilizadas como solução inicial pelo método de replanejamento.	58
Figura 35 - Comparação entre GRASP Reativo + Busca Tabu Estático x RepGuloso e RepGR.	59
Figura 36 - Comparação entre médias do resultado obtido no caso estático e no caso dinâmico (método de replanejamento).....	60

LISTA DE TABELAS

Tabela 1 - Rota utilizada pelo veículo 1 e solicitações atendidas durante trajeto.....	16
Tabela 2 - Rota utilizada pelo veículo 2 e solicitações atendidas durante trajeto.....	16
Tabela 3 - Média dos valores de Melhor FO e FO Médio e percentual de instâncias com melhor FO e melhor FO Médio obtidas considerando apenas Estratégia e Movimento.	45
Tabela 4 - Resumo dos resultados de FO Médio obtidos pelas estratégias e movimentos.	48
Tabela 5 - Média dos valores de FO Médio considerando apenas Frota e Capacidade.	49
Tabela 6 - Resultado da heurística Construtiva Gulosa (GOMES <i>et al.</i> , 2010).	51
Tabela 7 - Resultado da heurística GRASP-like (GOMES <i>et al.</i> , 2010).....	51
Tabela 8 - Resultado do método GRASP Reativo proposto.	51
Tabela 9 - Melhor Resultado Encontrado de cada instância para o caso estático sob os critérios de Melhor FO e FO Médio (não considerando GRASP Reativo + Busca Tabu).....	52
Tabela 10 - Resultado do método GRASP Reativo + Busca Tabu proposto.....	53
Tabela 11 - Resultado do replanejamento proposto com 20 solicitações a <i>posteriori</i> e solução inicial do GRASP Reativo + Busca Tabu proposto utilizando FO Médio.	55
Tabela 12 - Resultado do replanejamento proposto com 40 solicitações a <i>posteriori</i> e solução inicial do GRASP Reativo + Busca Tabu proposto utilizando FO Médio.	55
Tabela 13 - Média do resultado obtido pelos métodos de replanejamento considerando quantidade de solicitações posteriores e tipo de heurística utilizada como solução inicial.	57
Tabela 14 - Médias do resultado no caso estático e no caso dinâmico (método de replanejamento).	60
Tabela 15 - Melhor FO encontrado utilizando estratégia Primeiro Melhor.....	75
Tabela 16 - FO Médio utilizando estratégia Primeiro Melhor.....	75
Tabela 17 - Melhor FO encontrado utilizando estratégia Melhor Vizinho.....	75
Tabela 18 - FO Médio utilizando estratégia Melhor Vizinho.....	76
Tabela 19 - Resultado do replanejamento proposto com 20 solicitações a <i>posteriori</i> e solução inicial da heurística Construtiva Gulosa (GOMES <i>et al.</i> , 2010) utilizando FO Médio.....	76
Tabela 20 - Resultado do replanejamento proposto com 40 solicitações a <i>posteriori</i> e solução inicial da heurística Construtiva Gulosa (GOMES <i>et al.</i> , 2010) utilizando FO Médio.....	76

Tabela 21 - Resultado do replanejamento proposto com 20 solicitações a <i>posteriori</i> e solução inicial da heurística GRASP-like (GOMES <i>et al.</i> , 2010) utilizando FO Médio.....	76
Tabela 22 - Resultado do replanejamento proposto com 40 solicitações a <i>posteriori</i> e solução inicial da heurística GRASP-like (GOMES <i>et al.</i> , 2010) utilizando FO Médio.....	77
Tabela 23 - Resultado do replanejamento proposto com 20 solicitações a <i>posteriori</i> e solução inicial do GRASP Reativo proposto utilizando FO Médio.	77
Tabela 24 - Resultado do replanejamento proposto com 40 solicitações a <i>posteriori</i> e solução inicial do GRASP Reativo proposto utilizando FO Médio.	77

LISTA DE ABREVIATURAS E SIGLAS

BL – Busca Local

BT – Busca Tabu

DARP – *Dial-A-Ride Problem*

DRT – *Demand Responsive Transport*

DVRDRT – *Dynamic Vehicle Routing for Demand Responsive Transportation*

FO – Função Objetivo

GRASP – *Greedy Randomized Adaptive Search Procedure*

LC – Lista de Candidatos

LCR – Lista de Candidatos Restrita

NRF – Função de Ranqueamento de Nó

PGA – Problema Generalizado de Atribuição

PRV – Problema de Roteamento de Veículos

PRVD – Problema de Roteamento de Veículos Dinâmico

PRVDTRD – Problema de Roteamento de Veículos Dinâmico para Transporte
Reativo a Demanda

PRVJT – Problema de Roteamento de Veículos *Just-in-Time*

TRD – Transporte Reativo a Demanda

VRP – *Vehicle Routing Problem*

RESUMO

MIRANDA, Dângelo Silva, M.Sc., Universidade Federal de Viçosa, Agosto de 2012.

APLICAÇÃO DE METAHEURÍSTICAS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS DINÂMICO PARA TRANSPORTE REATIVO A DEMANDA. Orientador: André Gustavo dos Santos. Co-orientador: José Elias Cláudio Arroyo.

Os problemas de roteamento de veículos são de grande importância na pesquisa operacional, uma vez que envolvem problemas relacionados a logística de distribuição de mercadoria e/ou transportes de pessoas e possuem aplicabilidade em diversas áreas da economia. Este trabalho aborda o problema de roteamento de veículos estático e dinâmico para transporte reativo a demanda. Por ser um transporte sob demanda, as necessidades dos passageiros devem ser consideradas também durante a determinação das rotas dos veículos a fim de garantir a qualidade do serviço prestado, flexibilidade das rotas e eficiência na utilização do veículo com um mínimo de adição de ônus para o prestador. Visto que o problema possui características combinatórias, foi proposta para o caso estático uma heurística GRASP Reativo onde o parâmetro de aleatoriedade utilizado durante a construção da solução tem sua probabilidade de escolha autocalibrada. Foram propostas e testadas combinações de estratégias e movimentos a fim de determinar a escolha mais adequada para solução de instâncias do problema. Além disso, foi proposto um algoritmo híbrido substituindo-se a Busca Local do GRASP Reativo pelo método Busca Tabu. Por fim, foram propostos e testados algoritmos de replanejamento para o caso dinâmico. Os resultados obtidos pelos algoritmos propostos são comparados entre si e com resultados gerados por algoritmos da literatura. A análise e discussão dos resultados mostram que os métodos propostos alcançaram resultados satisfatórios, visto que as metaheurísticas obtiveram em média resultados pelo menos 10,27% melhores que algoritmos da literatura no caso estático, e os métodos de replanejamento conseguem melhorar ainda mais estes resultados.

ABSTRACT

MIRANDA, Dângelo Silva, M.Sc., Universidade Federal de Viçosa, August, 2012.
METAHEURISTICS APPLICATION TO THE DYNAMIC VEHICLE ROUTING FOR DEMAND RESPONSIVE TRANSPORTATION PROBLEM. Adviser: André Gustavo dos Santos. Co-Adviser: José Elias Cláudio Arroyo.

Vehicle routing problems have great interest in operations research, as they are logistics problems related to the delivery of goods and people transportation, and because they have applications in several fields of economy. This work deals with static and dynamic vehicle routing problem for demand responsive transportation. Being a responsive transportation, passengers' needs must be considered also along the route planning, in order to assure the quality or service, route flexibility and vehicle usage effectiveness, with minimum impact in the costs for the company. Since the problem has combinatorial characteristics, a Reactive GRASP heuristic is proposed for the static case. The randomness parameter used in construction phase has its probability self-adjusted. To find the approach best suited for the set of instances, different combinations of strategies and neighborhoods are proposed and tested. Besides that, it is proposed an hybrid algorithm, substituting the local search of the Reactive GRASP by an Tabu Search method. Furthermore, dynamic replanning algorithms were proposed and tested for the dynamic scenario. The results of the proposed methods are compared one to each other and to results of algorithms from the literature. The experimental analisys shows that the proposed methods give satisfactory results, as the metaheuristics obtained in average solutions 10,27% better than those from the literature in the static scenario, and the dynamic replanning methods improve the results even further.

1 INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV), que em inglês é *Vehicle Routing Problem* (VRP), é o nome genérico atribuído a uma classe de problemas de distribuição (SOSA *et al.*, 2007), cujo objetivo é determinar rotas de entrega e/ou coleta que minimizem o custo total do conjunto das rotas da frota, satisfazendo às restrições, de tempo, tamanho da frota, demanda, etc., que podem ser conhecidas antecipadamente ou não.

A otimização deste problema faz-se necessária devido a sua alta aplicabilidade prática, o que impacta diversas áreas da economia: indústrias, comércio, transporte, setor de serviços, lazer, segurança, etc. Além disso, a distribuição física dos produtos pode contribuir com cerca de 16% do custo final do item (BODIN *et al.*, 1983) *apud* (GOLDBARG; LUNA, 2005), sendo que na maioria das vezes a distribuição deverá ser eficaz e segura.

Algumas aplicações objeto de estudo são: coleta e entrega de passageiros de uma empresa (RODRIGUES, 2008), gerenciamento de frota, companhias de resgate e reparo, serviços de emergência, serviços de táxi, serviços de coleta e entrega de mercadorias, e até mesmo em redes de computadores (SANTOS, 2008), entre outros.

O PRV é um problema de natureza combinatória e pertence à classe NP-Difícil (LENSTRA; KAN, 1981). Algoritmos exatos são inviáveis para problemas de grande porte e há inúmeros trabalhos que utilizam métodos heurísticos/metaheurísticos para busca de solução: *Push-Forward Insertion Heuristic* (SILVA, 2008), Algoritmo Genético (PANKRATZ, 2005; RIBEIRO; LORENA, 2005), Busca Local (BENT; HENTENRYCK, 2004), Busca Tabu (BRÄYSY; GENDREAU, 2002; TAILLARD, E. *et al.*, 1997), *Greedy Randomized Adaptive Search Procedure* (KONTORAVDIS; BARD, 1995; LI *et al.*, 2004), *Simulated Annealing* (OLIVEIRA *et al.*, 2008), *Iterated Local Search* (SUBRAMANIAN *et al.*, 2008), Colônia de Formigas (RODRIGUES, 2008), Heurística Construtiva (PUREZA; LAZARIN, 2009; XAVIER *et al.*, 2008), *Variable Neighbourhood Search* (GOEL; GRUHN, 2008),

Large Neighbourhood Search (GOEL; GRUHN, 2008), *Reactive Variable Neighborhood Search* (BRAYSY, 2003), *Scatter Search* (SOSA *et al.*, 2007), *Particle Swarm* (MARINAKIS; MARINAKI, 2010; MARINAKIS *et al.*, 2010), entre outros.

Conforme (GHIANI *et al.*, 2003), “o PRV é estático se a entrada de dados (tempos de viagens, demandas, ...) não depende explicitamente do tempo, caso contrário ele é dinâmico. Além disso, um PRV é determinístico se todas as entradas de dados são conhecidas quando as rotas dos veículos são projetadas, caso contrário ele é estocástico”. Logo temos a seguinte classificação:

- **PRV estático:** o problema pode ser determinístico ou estocástico. No PRV **determinístico e estático**, todos os dados são conhecidos antecipadamente e o tempo não é levado em consideração; um exemplo é o clássico PRV capacitado (TOTH; VIGO, 2001). Já no PRV **estocástico e estático**, todas as rotas de veículos são projetadas no início do horizonte de planejamento, antes de dados incertos tornarem-se conhecidos; como exemplo temos PRV capacitado com demandas estocásticas.
- **PRV dinâmico:** o problema também pode ser determinístico ou estocástico. Nos problemas **determinísticos e dinâmicos**, todos os dados são conhecidos antecipadamente e alguns elementos de informação dependem do tempo; um exemplo é o PRV com janela de tempo revisto em (CORDEAU *et al.*, 2001) *apud* (GHIANI *et al.*, 2003). Por fim, temos os problemas **estocásticos e dinâmicos** (também conhecidos como problemas de envio e roteamento em tempo real). Como os dados são gradualmente revelados durante o intervalo de operação, as rotas não são construídas de antemão. Ao invés disso, as solicitações dos usuários são enviadas para os veículos à medida que os novos dados chegam; um exemplo deste tipo é mostrado em (PSARAFTIS, 1988).

1.1 Motivação

A busca por sobrevivência em um mercado competitivo, tem levado empresas de transportes públicos não apenas a reduzir seu quadro de funcionários e/ou

investimento em frota, mas também a limitar ou até mesmo a não disponibilizar o serviço em regiões de demanda baixa, variável e imprevisível como, por exemplo, em áreas rurais ou durante alguns períodos do dia em áreas urbanas.

A escassez de um transporte que atenda as necessidades dos usuários impulsiona a solução deste problema, na maioria das vezes, individualmente (aquisição de veículo), ocasionando aumento da frota de veículos em circulação e sobrecarga do sistema viário. Além disso, há o desperdício de tempo e combustível ao longo do trajeto, subutilização do veículo (lotação abaixo da capacidade), aumento da poluição ambiental (monóxido de carbono) e sonora, aumento dos gastos com manutenção, nível de stress, risco de acidentes, dentre outros.

Como acréscimo à situação mencionada, temos a carência que a gestão de transportes públicos possui de ferramentas que auxiliem no planejamento adequado dos recursos, uma vez que há uma grande diversidade de cenários a serem atendidos e um alto custo para prover o serviço nestas regiões.

Um dos problemas conhecidos dentro da gestão de transportes é a dificuldade em determinar o melhor conjunto de rotas de veículos em que a distância total percorrida pela frota seja mínima e ainda consiga atender às restrições impostas que podem ser horário de operação, demanda de passageiros pelo serviço, capacidade de cada veículo, dentre outros. Esse problema caracteriza-se como Problema de Roteamento de Veículos (PRV). Na maioria das situações reais, algumas ou todas as informações necessárias para a tomada de decisão sofrem alteração ao longo desse processo, isto é, possuem características dinâmicas. Logo, dizemos ser uma variante do PRV chamada de PRV Dinâmico (PRVD) (GOLDBARG; LUNA, 2005).

Além deste problema de gestão ser um PRVD, ele possui objetivos conflitantes entre si que são satisfazer as necessidades do passageiro e empresa. Logo, encontrar uma solução ótima é considerado um grande desafio. Porém, há na literatura heurísticas para geração de soluções que atendem os objetivos buscados (minimizar ou maximizar).

Neste trabalho é abordado um Problema de Roteamento de Veículos Dinâmico para Transporte Reativo a Demanda (PRVDTRD), onde passageiros especificam solicitações de transporte com janelas de tempo para embarque e desembarque, devendo obrigatoriamente embarcar no horário especificado. Para realização do atendimento, uma quantidade limitada de veículos sai do depósito (garagem), percorrendo os pontos de parada dentro das especificações solicitadas, podendo

ocorrer embarque e desembarque simultâneos, retornando à garagem ao fim da operação. Um dos primeiros trabalhos propostos para o PRVDTRD foi apresentado em (GOMES *et al.*, 2010). Neste foi exposto a descrição do problema, uma proposta incipiente de modelagem matemática e a aplicação de métodos heurísticos para minimizar a quantidade de não atendimentos, distância total percorrida e quantidade de veículos utilizada. Além disso, provaram que o mesmo era NP-Difícil simplesmente ao reduzi-lo a PRV clássico. Logo, a fim de obtermos soluções de boa qualidade para casos do PRVDTRD de tamanho real o uso de heurísticas torna-se interessante.

Portanto, a complexidade para obtenção de soluções de boa qualidade em tempo computacional razoável e a importância do problema proporcionam a motivação para a pesquisa fundamentada no uso de métodos heurísticos.

1.2 Objetivos

O objetivo geral do trabalho é estudar, compreender e descrever os comportamentos e resultados obtidos na aplicação da metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) Reativo e um método híbrido combinando a metaheurística com busca tabu para resolução do PRVDTRD estático, além de heurísticas para replanejamento no caso dinâmico, de forma a minimizar a distância total percorrida, a quantidade de solicitações não atendidas e os impactos de um replanejamento.

Os objetivos específicos desse trabalho são:

- Realizar uma revisão bibliográfica a respeito das características do problema e ferramentas disponíveis para ele;
- Buscar ou propor um conjunto de casos do problema com características comuns;
- Implementar heurísticas propostas na literatura a fim de possibilitar comparação com as metaheurísticas propostas neste trabalho;
- Implementar estratégias e movimentos (pequenas alterações de estrutura na solução) que possibilitem melhorar a qualidade da solução final encontrada;
- Aplicar a metaheurística GRASP Reativo e seu híbrido com Busca Tabu e analisar os resultados apresentados;
- Propor e aplicar heurísticas de replanejamento para o caso dinâmico;

- Comparar o desempenho das estratégias e movimentos a fim de determinar qual é a mais adequada ao problema estático;
- Comparar os desempenhos das metaheurísticas desenvolvidas neste trabalho com o desempenho de heurísticas propostas na literatura;
- Comparar o desempenho das heurísticas de replanejamento para o caso dinâmico com os obtidos no caso estático;
- Analisar os resultados encontrados, visando determinar padrões que melhor se adaptem a situação estática e dinâmica do problema;
- Apresentar os resultados alcançados que comprovem a viabilidade da utilização de metaheurísticas híbridas e de replanejamento, assim como as vantagens que podem ser alcançadas na resolução de problemas reais.

1.3 Organização do documento

O Capítulo 2 apresenta uma breve classificação dos problemas associados a roteamento de veículos, os conceitos associados a Transporte Reativo a Demanda e trabalhos relacionados ao assunto.

O Capítulo 3 apresenta uma definição detalhada do problema, suas particularidades em relação aos outros da classe de problemas de roteamento de veículos.

No Capítulo 4 são descritos as metaheurísticas e métodos de replanejamento propostos, além da forma adotada de representar a solução nestes métodos e procedimentos de melhoria.

O Capítulo 5 expõe todos os detalhes necessários para realização dos testes computacionais, além dos resultados alcançados e a análise e discussão destes.

O Capítulo 6 apresenta as conclusões e objetivos alcançados a partir da realização dos testes computacionais e inclusive propostas de trabalhos a serem realizados futuramente.

Ao final deste documento são apresentadas referências bibliográficas utilizadas para embasamento teórico deste trabalho.

2 REFERENCIAL TEÓRICO

Como o Problema de Roteamento de Veículos (PRV) tem sido bastante estudado nas últimas décadas, atualmente há um aumento no interesse por logísticas baseadas no conceito *just-in-time*, isto é, de reações rápidas, norteadas pela informação em tempo real ou sob demanda, e os avanços nas áreas de hardware e telecomunicações tem disponibilizado uma grande quantidade de informações em tempo real que podem orientar o processo de gestão de transportes e distribuição mais eficiente.

Diante disso deve-se levar em consideração as características do problema que está sendo tratado e o grau de dinamismo e incerteza associado às informações necessárias para resolução.

Para tanto, é necessário identificar as características preponderantes do problema, classificá-lo e então determinar as abordagens de resolução que mais se adaptam a ele.

Primeiramente é apresentado, de maneira breve, classificações propostas para os PRVs em relação ao momento em que as informações necessárias para resolução do problema estão disponíveis e também se podem sofrer alterações ao longo do tempo.

Em seguida, o conceito de Transporte Reativo a Demanda (TRD) é exposto a fim de ampliar o entendimento sobre a flexibilidade envolvida no planejamento da rota e horários de veículos, bem como os compromissos necessários para seu funcionamento e vantagens de sua utilização.

Por fim, são comentados resumidamente alguns trabalhos relacionados ao PRV Dinâmico e as abordagens utilizadas nestes.

Neste capítulo é apresentado uma breve classificação para os PRVs, o conceito de Transporte Reativo a Demanda e alguns trabalhos relacionados.

2.1 Classificação do problema

Dada a diversidade de Problemas de Roteamento de Veículos (PRV) e a dificuldade em classificá-los, é sugerido em (MADSEN *et al.*, 2007) a classificação em Estático e Dinâmico. Estático quando todas as informações relevantes para o planejamento das rotas são conhecidas quando o planejador inicia o processo de roteamento; e estas informações não são dependentes do tempo, isto é, não sofrem alteração ao longo do tempo após a construção das rotas. Já no Dinâmico, as informações são dependentes do tempo e conhecidas *a posteriori* ao planejamento das rotas. As solicitações podem ser do tipo Antecipadas (chegam antes do processo de roteamento iniciar) ou Imediatas (aparecem em tempo real durante a execução das rotas). O grau de dinamismo depende não apenas da quantidade de consumidores e o espaço de distribuição, mas também da quantidade de eventos dinâmicos e o momento em que eles ocorrem. Apesar de algumas vezes os objetivos no PRV Dinâmico (PRVD) serem conflitantes, alguns elementos são frequentemente relevantes ao definir o objetivo e por isso, os objetivos podem ser classificados em: minimização dos custos da distribuição, maximização do nível de serviço, maximização da “vazão” (quantidade de veículos roteados em um determinado espaço de tempo). Por fim, é proposto uma classificação em níveis que distingue entre sistemas fraco, moderado e fortemente dinâmico.

Como mencionado anteriormente, em (GHIANI *et al.*, 2003) os PRVs são classificados em: estático ou dinâmico, determinístico ou estocástico. “O PRV é estático se a entrada de dados (tempos de viagens, demandas, ...) não depende explicitamente do tempo, caso contrário ele é dinâmico. Além disso, um PRV é determinístico se todas as entradas de dados são conhecidas quando as rotas dos veículos são projetadas, caso contrário ele é estocástico”. Ademais, é discutido a respeito do tempo de resposta, negação de serviço, congestionamento, o grau de dinamismo e sua relação com os objetivos do problema, alguns algoritmos sequenciais, estratégias de paralelização computacional, e finalmente, assuntos gerais e específicos de PRVD que podem ser desenvolvidos.

2.2 Transporte Reativo a Demanda

Transporte Reativo a Demanda (TRD), que em inglês é *Demand Responsive Transport* (DRT), é um serviço de transporte sob demanda, perto das necessidades

dos passageiros, uma vez que há flexibilidade das rotas e horários dos veículos. Os passageiros determinam os locais e horários de embarque e desembarque (AMBROSINO *et al.*, 2003), além de outros aspectos a serem considerados.

Em (JOSSELIN *et al.*, 2008), um TRD Dinâmico seria um possível consenso/compromisso entre Flexibilidade, Qualidade de Serviço e Eficiência. A Flexibilidade ocorre uma vez que desvios de rota inesperados poderão ocorrer a fim de atender a solicitação recém-chegada de um passageiro ao sistema.

A Figura 1 ilustra uma situação em que seria necessário apenas uma rota para atendimento dos passageiros conhecidos pelo sistema e duas solicitações acabam de chegar. Neste momento o conceito de Flexibilidade é aplicado: determina-se um desvio na rota elaborada para atendimento a um dos passageiros recém-conhecidos e a criação de outra rota para dedicar-se à solicitação restante.

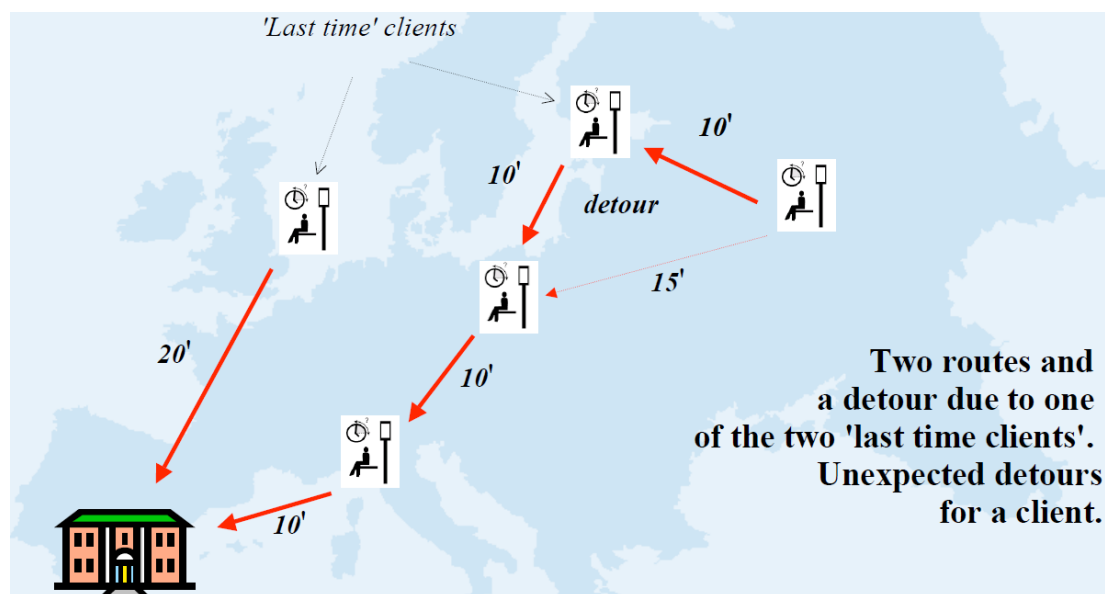


Figura 1 - Exemplo do critério Flexibilidade. Fonte: (JOSSELIN *et al.*, 2008).

Qualidade de Serviço assegura o atendimento de necessidades do passageiro durante a elaboração da rota: minimizar tempo a bordo, minimizar tempo de espera por atendimento, dentre outros.

A Figura 2 demonstra que para assegurar a qualidade de serviço (por exemplo o tempo a bordo não exceder em 35 minutos), pode não ser possível utilizar desvios para o atendimento dos passageiros recém-conhecidos (Figura 1). Como consequência, mantém-se a rota pré-definida que atende os passageiros agendados e inicia-se outra para as novas solicitações.

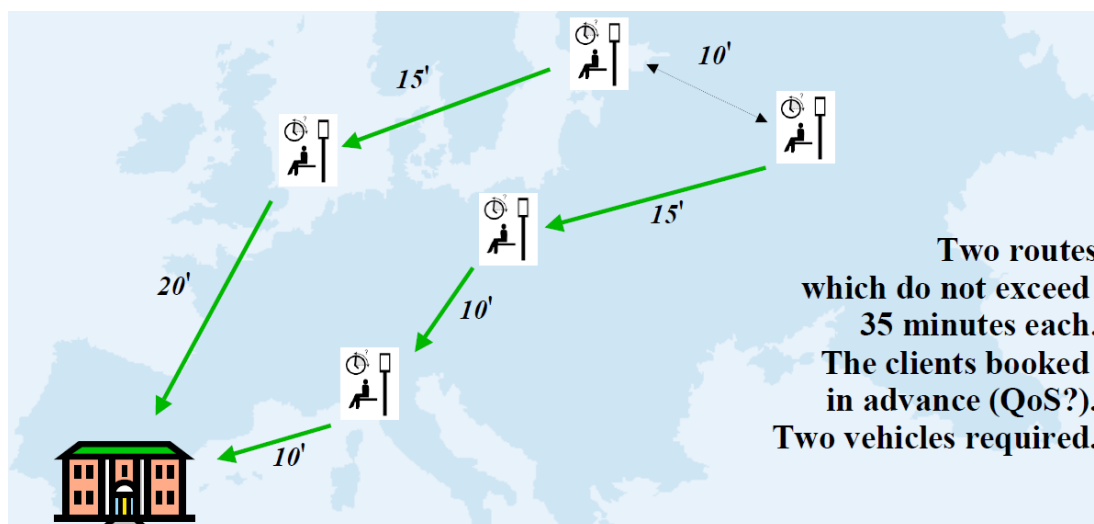


Figura 2 - Exemplo do critério Qualidade de Serviço. Fonte: (JOSSELIN *et al.*, 2008).

Por fim, Eficiência garante maximizar o atendimento das solicitações com um mínimo de veículos possíveis, respeitando as restrições de capacidade e não se preocupando com o nível de serviço aceitável para o passageiro.

A Figura 3 expõe a situação de Eficiência, onde a rota pré-definida com passageiros agendados teve a inclusão de desvios para atendimento das solicitações recém-conhecidas (Figura 1), o que possibilitou utilizar apenas um único veículo para resolver a situação. Entretanto, isso aumenta o tempo de viagem total para 55 minutos.

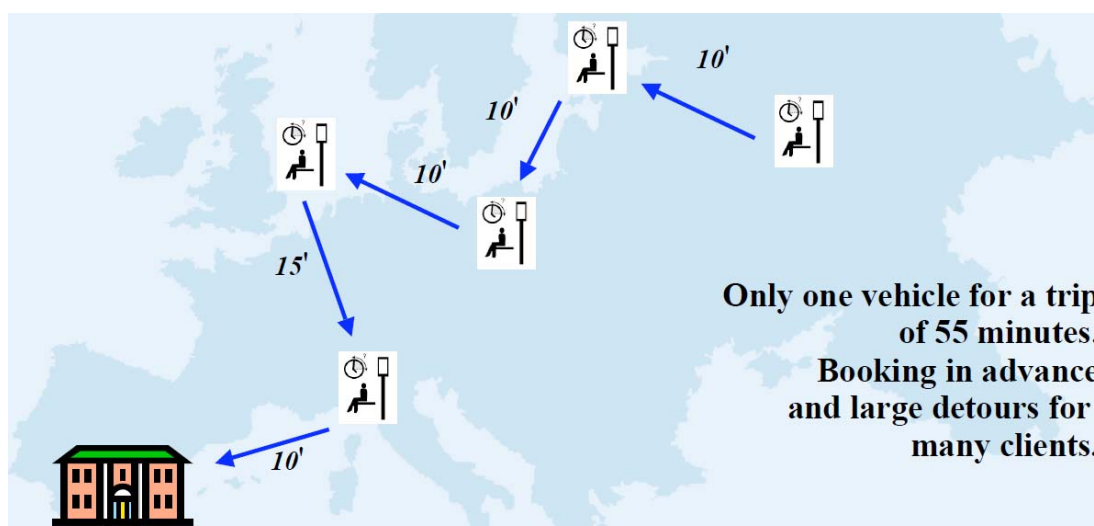


Figura 3 - Exemplo do critério Eficiência. Fonte: (JOSSELIN *et al.*, 2008).

No trabalho (JOKINEN *et al.*, 2011) são apresentados como resultados das simulações, algumas vantagens observadas da utilização de transporte urbano

baseado em TRD: é um modal de transporte que possui um custo por viagem menor que outros modais (carro, taxi) (Figura 4), possui a capacidade de tolerar mudanças súbitas inesperadas na demanda sem qualquer mudanças nos parâmetros de operação (*resilient*) (Figura 5), redução do nível de stress relacionado a dirigir e estacionar, dentre outros.

A Figura 4 mostra que à medida que ocorre um aumento na quantidade de viagens por dia, o custo operacional por viagem é diluído.

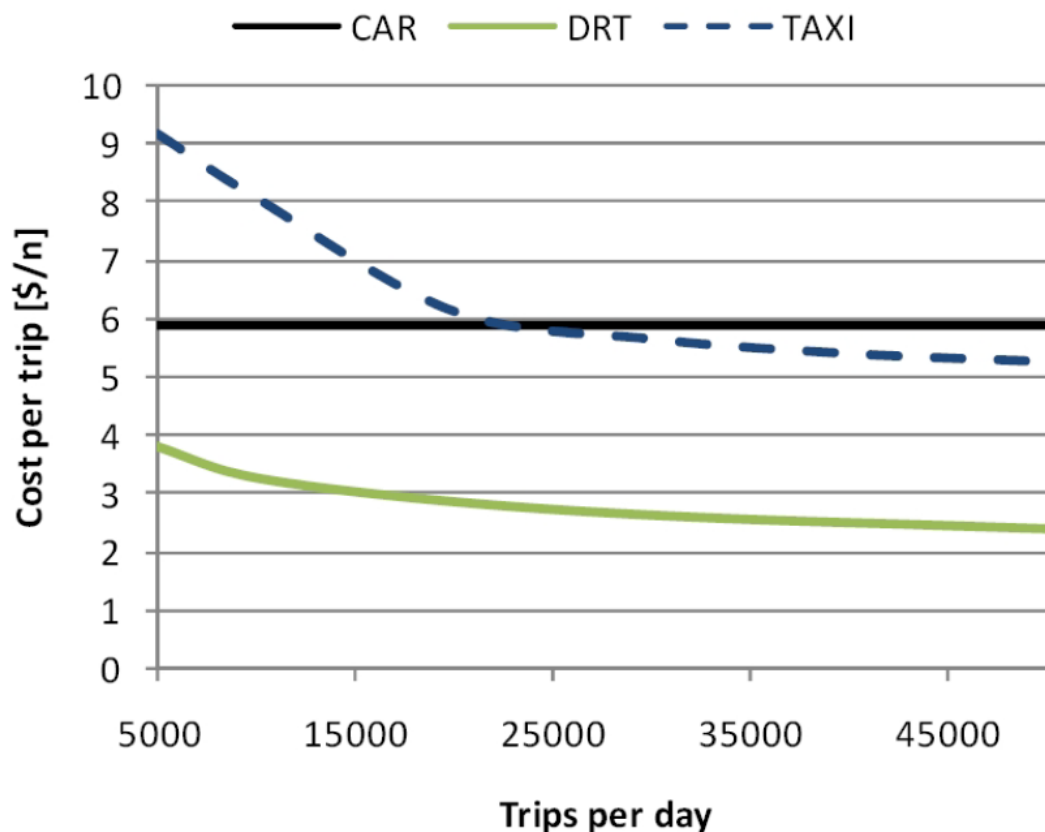


Figura 4 - Estimativa do custo médio por viagem em função das viagens por dia.
Fonte: (JOKINEN *et al.*, 2011).

A Figura 5 mostra a densidade demandada ao iniciar o serviço (30 viagens/Km²/hora) e a média do tempo percorrido. À medida que surgem solicitações inesperadas, isto é, ocorre um aumento na densidade demandada, o modal táxi, rapidamente sofre um aumento considerável na média do tempo percorrido, resultando em uma demora na prestação do serviço.

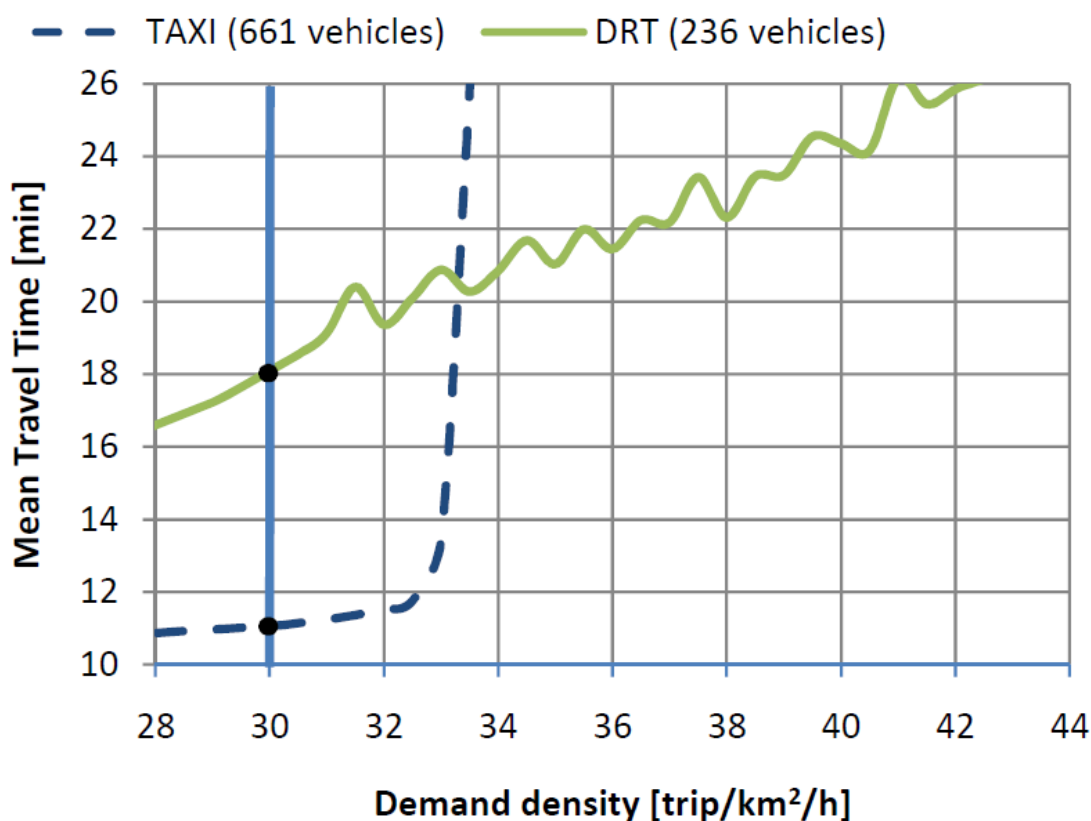


Figura 5 - Sistema TRD é "resilient". Fonte: (JOKINEN *et al.*, 2011).

2.3 Trabalhos Relacionados

Alguns trabalhos que estão relacionados com o Problema de Roteamento de Veículos Dinâmico (PRVD) e suas variações estão referenciados a seguir.

Em (FLEISCHMANN *et al.*, 2004) é apresentado um framework de planejamento para o PRVD onde compara três procedimentos de roteamento para envio baseado em eventos, sendo que a diferença entre eles é o comprimento do horizonte de planejamento por evento. Como o objetivo é a utilização de informações de tempo de viagem dinâmico, foi utilizado o cálculo do menor caminho dinâmico. Por fim, os procedimentos foram testados e comparados utilizando dados reais de um centro de administração de tráfego urbano e um provedor de serviço logístico.

O trabalho (PUREZA; LAZARIN, 2009) avalia o impacto da incorporação de técnicas para PRVD em ambientes de prestação de serviços, onde o objetivo prioritário é minimizar o prazo de atendimento. Para isso, utiliza uma heurística construtiva/desconstrutiva baseada na Heurística *i* (SOLOMON, 1987) para o PRVJT, elaborando rotas em tempo real ao longo de um horizonte de planejamento. O algoritmo proposto foi testado e comparado utilizando conjuntos de instâncias

geradas aleatoriamente baseadas em dados fornecidos por uma empresa do setor de bebidas do interior do estado de São Paulo.

Utilizando o Problema Generalizado de Atribuição – PGA (FISHER; JAIKUMAR, 1981) com aplicação de relaxação e reformulação do PGA em problema de cobertura de conjuntos e resolvido através da aplicação do algoritmo do caixeiro viajante, foi desenvolvido em (CALDAS; SANTOS, 2008) a adequação de alternativas de roteamento de veículos em relação a dois cenários simulados levando em consideração o posicionamento geográfico dos pontos de venda em relação a uma central de distribuição de bebidas. Para isso, foram consideradas 4 alternativas de roteamento: uma baseada no algoritmo de (FISHER; JAIKUMAR, 1981) priorizando a formação de grupos de pontos de venda (clusterização); e o restante, baseadas em variações crescentes da possibilidade de inserção e redução de rotas (r-opt). Todos os algoritmos foram submetidos a cenários onde os pontos de venda concentrados em uma determinada região geográfica podem ou não estarem próximos uns dos outros.

É apresentado em (MONTEMANNI *et al.*, 2003) um algoritmo colônia de formigas baseado na decomposição do PRVD em uma sequência de PRVs estáticos. O recebimento de novos pedidos, a sequência de pedidos já atendidos, a posição e a capacidade residual de cada veículo são informações utilizadas para construção do PRV estático. O dia de trabalho é dividido em fatias de tempo, a qual é associada um PRV estático que considera todos os pedidos já recebidos que ainda não foram executados. Os 21 “benchmarks” adotados são derivados do proposto em (KILBY *et al.*, 1998), os quais são baseados em “benchmarks” de PRV estáticos muito populares, originalmente utilizados em (CHRISTOFIDES; BEASLEY, 1984; FISHER; JAIKUMAR, 1981; TAILLARD, É., 1993). Finalmente, o algoritmo proposto é testado em um caso de estudo real, configurado na cidade de Lugano, Canton Ticino, Suíça.

Em (GOMES *et al.*, 2010) é apresentada a modelagem matemática de alguns objetivos do PRVDTRD (em inglês, *Dynamic Vehicle Routing for Demand Responsive Transportation – DVRDRT*) inspirado em (CORDEAU; LAPORTE, 2007; CORDEAU *et al.*, 2005; CORDEAU; LAPORTE; POTVIN, J. Y. *et al.*, 2007; CORDEAU; LAPORTE; SAVELSBERGH; *et al.*, 2007; MACHADO *et al.*, 2002; PSARAFTIS, 1988, 1995). Entretanto, os autores não propõem uma formulação matemática completa, uma vez que as restrições do problema não são definidas da mesma forma. O trabalho também apresenta uma abordagem heurística, propondo

um algoritmo construtivo guloso e uma heurística melhorada randomizada inspirada no GRASP, com respectivas soluções para um exemplo hipotético. Por fim, é analisada a eficiência dos algoritmos em um conjunto de instâncias geradas aleatoriamente.

Recentemente foi proposto um modelo matemático para solução de uma variação do PRVDTRD aqui estudado (HERBAWI; WEBER, 2012). Os autores estudam um problema chamado *ridematching problem with time windows in dynamic ridesharing*, em que um conjunto de veículos, cada um com horário e local de partida e chegada, podem ser usados para transportar (carona) outras pessoas localizadas na região. Diferentemente do trabalho aqui estudado, os veículos não partem do mesmo local, e janelas de tempo são tratados de forma diferente. Porém, é possível adaptar o modelo, embora não seja viável sua utilização para problemas reais por causa da complexidade de solução exata.

3 DEFINIÇÃO DO PROBLEMA

Por ser uma variação do PRVD, o Problema de Roteamento de Veículos Dinâmico para Transporte Reativo a Demanda (PRVDTRD) é desafiante não apenas por causa da natureza combinatória, mas devido a necessidade de disponibilizar um roteamento flexível, preparado para as mudanças ao longo do tempo, e simultaneamente não degrade o nível do serviço prestado nem cause um aumento nos custos. Desta forma, busca-se tanto maximizar o nível de atendimento quanto minimizar os custos envolvidos. Neste trabalho são utilizadas a definição e as características do problema propostas em (GOMES *et al.*, 2010).

Inicialmente é apresentado a definição do PRVDTRD conforme (GOMES *et al.*, 2010). Em seguida, um exemplo hipotético com todos os dados necessários e uma solução para este exemplo. O exemplo mostrado trata apenas o caso estático, sendo o caso dinâmico tratado no capítulo seguinte.

3.1 Definição do Problema

Conforme mencionado no trabalho (GOMES *et al.*, 2010), no Problema de Roteamento de Veículos Dinâmico para Transporte Reativo a Demanda (PRVDTRD) as rotas e frequências podem variar de acordo com a demanda observada (solicitações de transporte recebidas em qualquer momento que possuem origem e destino, com as suas respectivas janelas de tempo, definidas pelos passageiros). Cada vértice de parada pertencente a uma rota, exceto o depósito, aceita o embarque e desembarque simultâneo de passageiros, isto é, pode ter múltiplas janelas de tempo com possível sobreposição e por isso, as novas solicitações recebidas em tempo real podem exigir a visita neste em diferentes períodos, diferenciando das variações PRV e *Dial-A-Ride Problem* (DARP). A frota de veículos possui igual capacidade (número de assentos), com início e término das rotas no único depósito existente. Como possui múltiplos objetivos que conflitam entre si, o problema assemelha-se a um DARP multi-objetivo (BRANKE *et al.*, 2008), não sendo possível determinar a melhor solução, mas o conjunto de soluções

eficientes que mais se aproxima da fronteira de Pareto e satisfaz as necessidades desejadas: Qualidade de Serviço (minimizar o tempo de espera e tempo de viagem do passageiro), Flexibilidade (rotas flexíveis) e Eficiência (minimizar o custo da rota total e maximizar as solicitações atendidas).

O PRVDTRD tratado em (GOMES *et al.*, 2010) tem as seguintes características:

- Múltiplos veículos com igual capacidade;
- Único depósito onde os veículos iniciam e terminam as rotas;
- Embarque e desembarque simultâneos;
- Usuários especificam solicitações de transporte de qualquer lugar para qualquer lugar (*many-to-many*), em qualquer tempo (*dynamic*);
- Usuários especificam janelas de tempo para embarque e desembarque;
- Múltiplas janelas de tempo em cada parada (possivelmente sobrepostas);
- Janelas de tempo de embarque devem ser respeitadas (*hard constraint*);
- Janelas de tempo de desembarque podem ser violadas sob uma penalidade (*soft constraint*).

Neste problema, $P = \{1, 2, \dots, p\}$ representa o conjunto de solicitações de passageiros contendo todas as solicitações de transporte recebidas, sendo que p é o número de solicitações e cada passageiro corresponde a uma e somente uma solicitação. O problema pode ser modelado como um grafo formado por um conjunto de vértices e arestas. O conjunto de vértices de parada V possui o depósito (0), juntamente com o conjunto dos nós de embarque e desembarque. Associado a cada arco a_{ij} que conecta o vértice i ao j , $i \neq j$ e $i, j \in V$, existe um custo (distância) c_{ij} e tempo d_{ij} de viagem para atravessá-lo. Para cada $v \in V - \{0\}$, Pin_v e $Pout_v$ são conjuntos que armazenam, respectivamente, as solicitações com embarque e desembarque em v ; e m_v é o índice de lotação, a diferença entre o número de solicitações de Pin_v e $Pout_v$. Note que não há embarque ou desembarque de passageiros no depósito ($Pin_0 = Pout_0 = \emptyset$).

Cada solicitação $i \in P$ possui as seguintes informações: identificação, vértice de embarque, vértice de desembarque, janela de tempo de embarque e janela de tempo de desembarque. Por exemplo, na solicitação denotada por id1049_v1-v5: [00h06,00h09]-[01h21,01h24], sua identificação é id1049, vértice de embarque 1,

vértice de desembarque 5, janela de tempo de embarque [00h06,00h09] e janela de tempo de desembarque [01h21,01h24].

O conjunto de veículos $K=\{1, 2, \dots, k\}$, sendo k o número de veículos, todos com mesma quantidade de assentos Q , representa a frota disponível.

Para ilustrar graficamente o problema, considere a Figura 6 como estado inicial para o PRVDTRD estático. Neste exemplo há 12 vértices de parada, sendo 11 pontos possíveis para embarque e/ou desembarque e o vértice 0, o depósito. Existem 5 solicitações de transportes pertencentes ao conjunto P : id4, id1049, id1050, id48 e id1011. Analisando-se as solicitações do conjunto P , temos que o conjunto de nós de embarque é $\{1, 2, 5\}$ e o de desembarque, $\{1, 3, 5\}$. Portanto, o conjunto de vértices de parada V é $\{0\} \cup \{1, 2, 5\} \cup \{1, 3, 5\}$. São representadas ainda as distâncias e tempos de viagem dos trechos onde o veículo pode trafegar.

Considere que para solução deste problema, estejam disponíveis 2 veículos de capacidade 10. Neste exemplo, as solicitações dos passageiros são:

- id1049_v1-v5: [00h06,00h09]-[01h21,01h24]
- id1050_v1-v5: [00h07,00h10]-[01h21,01h24]
- id4_v1-v3: [00h07,00h10]-[00h25,00h28]
- id1011_v2-v5: [00h16,00h19]-[02h40,02h43]
- id48_v5-v1: [00h33,00h36]-[02h38,02h41]

As tabelas 1 e 2 contêm possíveis rotas a serem realizadas, respectivamente, pelos veículos 1 e 2 e seus atendimentos. Tais rotas são ilustradas na Figura 7.

Tabela 1 - Rota utilizada pelo veículo 1 e solicitações atendidas durante trajeto

Rota Veículo 1: 0 1 3 5 0		
Identificação da solicitação	Horário de embarque	Horário de desembarque
id1049	00h06	00h26
id1050	00h07	00h26
id4	00h07	00h17

Tabela 2 - Rota utilizada pelo veículo 2 e solicitações atendidas durante trajeto

Rota Veículo 2: 0 2 5 1 0		
Identificação da solicitação	Horário de embarque	Horário de desembarque
id1011	00h16	00h32
id48	00h33	00h45

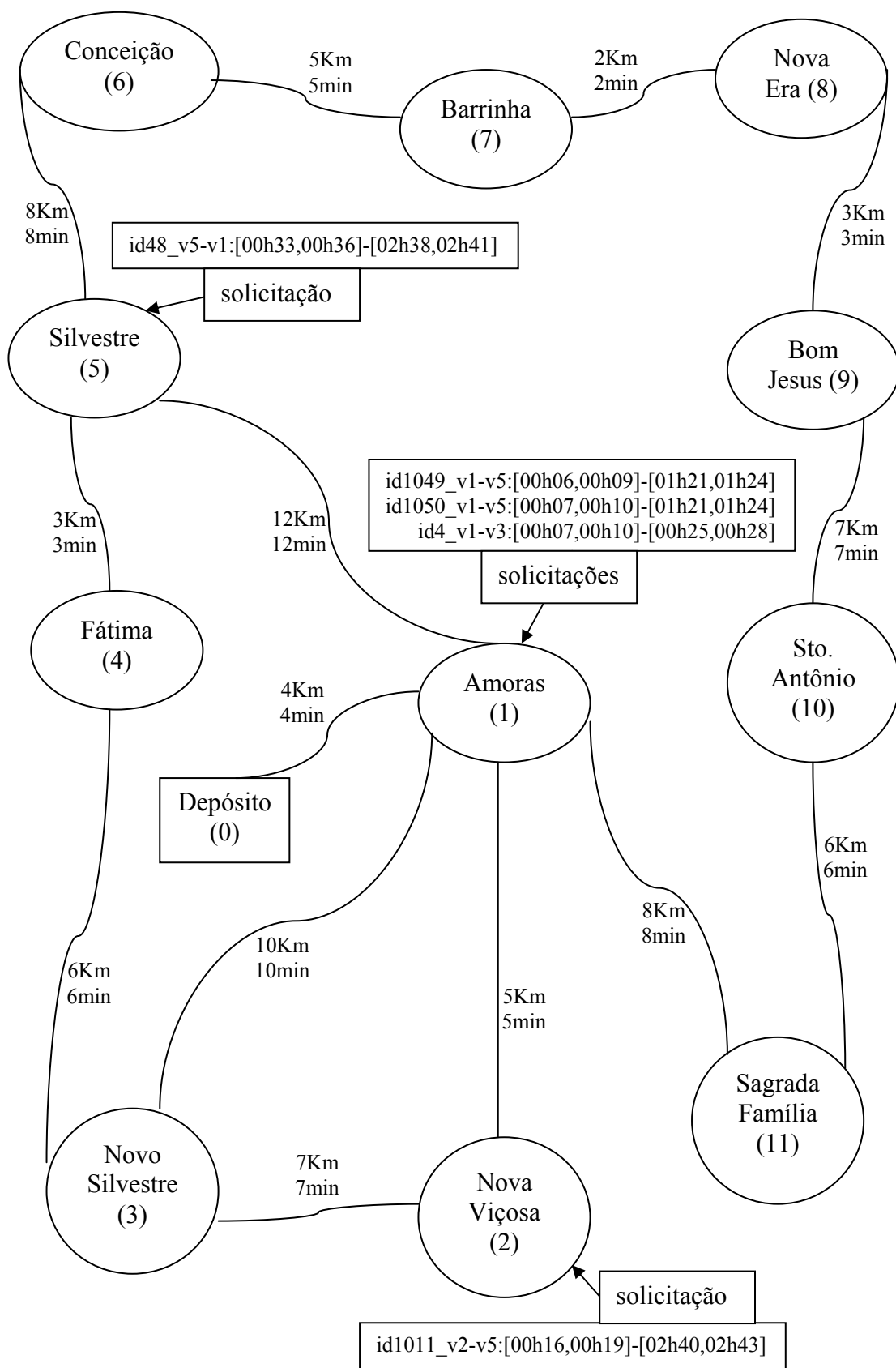


Figura 6 - Estado inicial hipotético para um PRVDTRD estático. Adaptado de (GOMES *et al.*, 2010).

O desenho da Figura 7 é o mesmo da Figura 6, porém com o indicativo das rotas que serão realizadas pelos veículos. A fim de ampliar o entendimento, lembre-se que o veículo realiza paradas exclusivamente em vértices que contenham solicitações de embarque e/ou desembarque, considerando os vértices restantes simplesmente como vértices de transbordo. Por isso, as setas representadas indicam apenas vértices de parada. Logo, para realizar o percurso entre estes vértices, o veículo utiliza sempre a rota com menor distância. Este percurso foi indicado apenas com o segmento de reta, podendo ligar vértices de transbordo ou apenas sobrepor-se a eles.

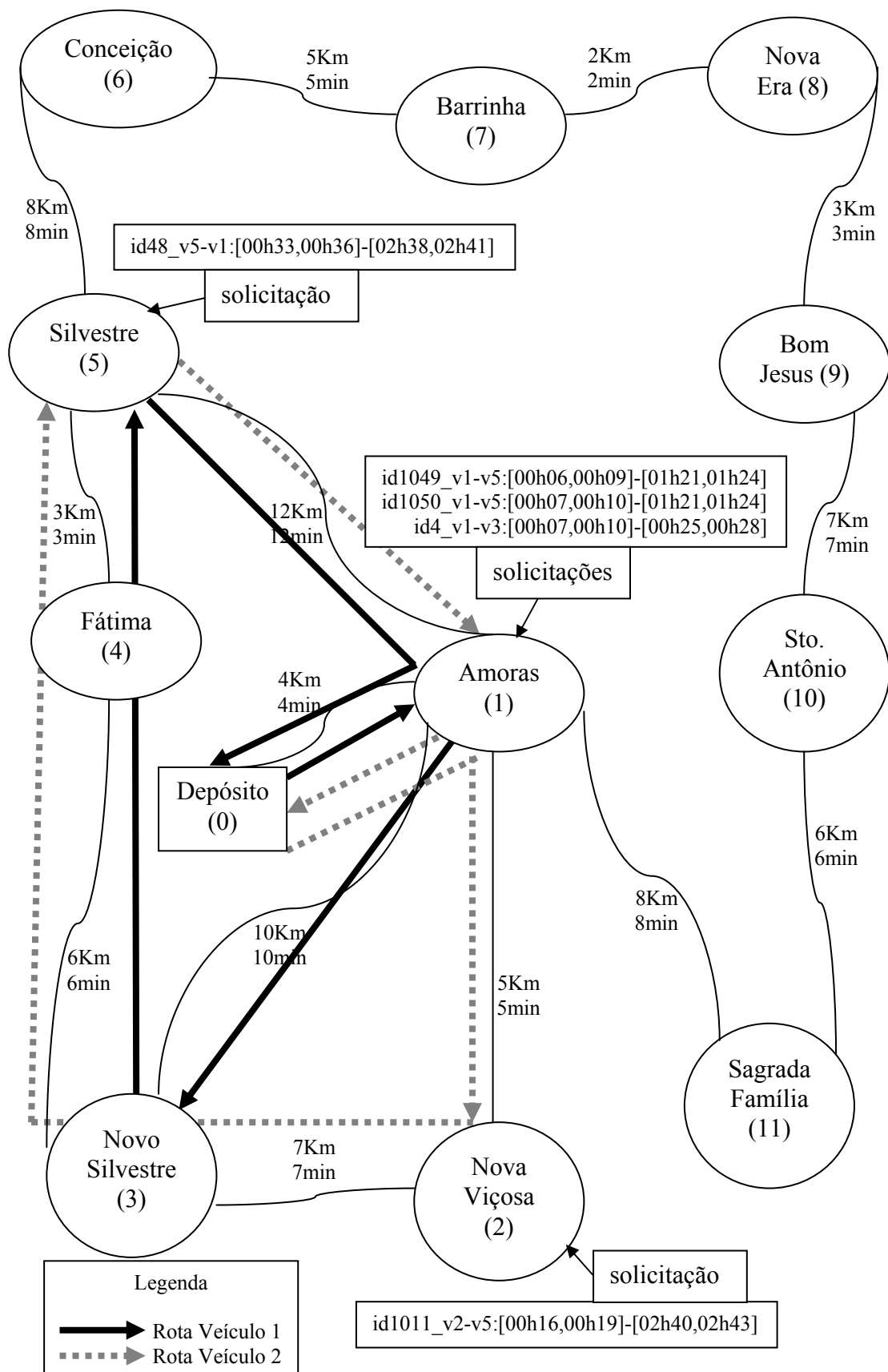


Figura 7 - Representação da solução para problema proposto na Figura 6.

4 MÉTODOS HEURÍSTICOS PROPOSTOS

Para resolução do PRVDTRD, encontram-se disponíveis na literatura os métodos Construtivo Guloso (*Greedy Constructive*) e GRASP-like propostos em (GOMES *et al.*, 2010). O Construtivo Guloso baseia-se exclusivamente na Função de Ranqueamento de Nó (*NRF*) (GOMES *et al.*, 2010) para construção de cada rota, onde a cada iteração do algoritmo é determinado o melhor nó naquele instante a ser inserido na rota segundo a ponderação das perspectivas do veículo (distância para todos os outros nós a partir da posição atual; quantidade de passageiros a ser atendido naqueles nós) e do passageiro (quantidade de passageiros a bordo do veículo com o mesmo local de desembarque; janela de tempo dos passageiros nos nós restantes). A função *NRF* é definida como sendo

Equação 1

$$NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i]), \forall i \in V - \{0\}$$

onde α_d , α_p , α_v , α_t são os pesos atribuídos aos critérios distância, número de passageiros, janela de tempo de desembarque e janela de tempo de embarque, respectivamente, e *CRL*, *NRL*, *DRL*, *TRL* são listas de ranqueamento do nó baseados no custo da viagem (distância) até o nó *i*, lotação do nó *i* (já explicado anteriormente), menor horário de desembarque naquele nó, menor horário de embarque naquele nó, respectivamente.

Já o algoritmo GRASP-like, a cada iteração escolhe probabilisticamente um parâmetro de aleatoriedade α em uma lista de possíveis parâmetros. Depois utiliza-se a função *NRF* para construir uma lista de candidatos (*LC*), ordenando decrescentemente esta lista por *NRF*[*i*]. Em seguida, utilizando o parâmetro de aleatoriedade α , constrói-se uma lista de candidatos restrita (*LCR*) contendo de 1 a no máximo $\alpha \times |LC|$, onde $|LC|$ é a quantidade de elementos em *LC*. Por ser um algoritmo reativo, a cada γ iterações ocorre uma auto-calibragem da probabilidade associada à lista dos possíveis parâmetros α que serão utilizados. Porém é importante ressaltar alguns detalhes: durante a construção de *LC* não é levada em consideração a viabilidade do elemento a ser adicionado e o algoritmo não utiliza uma fase de busca local para melhoria da solução encontrada.

Diante disso, neste trabalho foram propostos os seguintes métodos: GRASP Reativo, GRASP Reativo combinado com Busca Tabu, estratégias e movimentos a serem utilizados pela busca durante exploração e melhoria da solução encontrada. Além disso, foram propostos métodos para o caso dinâmico a fim de realizar o replanejamento das rotas com a chegada de novas solicitações.

Nas próximas subseções são detalhados não apenas estes métodos mas outras informações necessárias durante a construção dos mesmos.

Primeiramente são expostos os itens utilizados para representar uma solução do problema, desde os critérios para formação de uma rota e solução até o modo de avaliar o custo da rota, das solicitações não atendidas e da solução.

Depois é explicado o funcionamento do método GRASP, o processo utilizado durante a fase de construção de soluções viáveis e o mecanismo que o transforma em GRASP Reativo.

A Busca Local empregada no método GRASP é exposta em seguida. Nesta subseção são discutidos os tipos de movimento e estratégia desenvolvidos, bem como são exemplificados seus comportamentos.

A combinação entre os métodos GRASP Reativo e Busca Tabu é comentada, assim como a maneira que o procedimento consegue interferir na qualidade das soluções encontradas.

Em penúltimo lugar são descritos a Busca Tabu e seu funcionamento, além de explicar o algoritmo e exemplificar o comportamento da lista tabu.

Finalmente são apresentados e explicados os algoritmos de replanejamento utilizados no caso dinâmico do problema e as informações necessárias para utilização destes métodos.

4.1 Representação de uma solução do problema

Sendo k o número de veículos disponíveis na frota K , cada solução S é composta por um conjunto de n rotas R_i , isto é, $S = \{R_1, \dots, R_n\}$, $n \leq k$, formadas a partir do conjunto de vértices V que respeitam as restrições do problema: conservação de fluxo (todo veículo que entra em um nó deve deixá-lo); cada solicitação é atendida por um único veículo; cada passageiro deverá obrigatoriamente embarcar e preferencialmente desembarcar dentro de sua janela de tempo; cada solicitação do serviço de embarque deverá preceder a de desembarque.

Uma rota viável R_i é uma lista de vértices a qual deverá conter ao menos dois vértices além do depósito, correspondendo aos vértices de embarque e desembarque de pelo menos uma solicitação de transporte. Desta forma, uma rota mínima pode ser representada por $\min R = \langle 0, p^+, p^-, 0 \rangle$, onde p^+ e p^- representam os vértices de embarque e desembarque, respectivamente. Uma vez que é necessário respeitar as diversas restrições do problema, um veículo $l \in K$ poderá eventualmente retornar a um vértice já visitado para atendimento de alguma solicitação, porém a repetição do vértice não poderá ser consecutiva.

Cada rota $R_i = \langle v_0, v_1, \dots, v_m, v_{m+1} \rangle$ tem uma função custo

Equação 2

$$C(R_i) = \sum_{j=0}^m (c_{v_j, v_{j+1}}) + W$$

onde $v_0 = v_{m+1} = 0$ (depósito), $c_{v_j, v_{j+1}}$ é o custo para atravessar o arco (v_j, v_{j+1}) , e W é o custo fixo para utilização do veículo.

É importante ressaltar que para algumas instâncias é impossível atender todas as solicitações pelos seguintes motivos: janela de tempo estreita demais e/ou quantidade limitada de veículos disponíveis. Todas as solicitações que não foram atendidas fazem parte do conjunto $U = \{1, 2, \dots, u\}$. De forma análoga à rota, o custo associado a este conjunto é

Equação 3

$$C(U) = \sum_{j \in U} Y$$

onde Y é o custo fixo pelo não atendimento da solicitação.

Logo, o custo total da solução S é:

Equação 4

$$C(S) = \left(\sum_{i=1}^n C(R_i) + C(U) \right)$$

A Função Objetivo (FO) de uma solução S , a qual avalia a qualidade desta, foi a mesma utilizada no trabalho (GOMES *et al.*, 2010), isto é

Equação 5

$$FO(S) = C(S)$$

4.2 Metaheurística GRASP

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*), proposta em (FEO; RESENDE, 1995) é um método iterativo onde é criada uma

solução viável para o problema desejado utilizando um parâmetro de aleatoriedade (α) fixo seguido de um estágio de aperfeiçoamento e armazenando a melhor solução obtida até o momento. A iteração é composta pelas fases de construção e busca local, nesta sequência. Durante a primeira fase, uma lista de candidatos (LC) ordenada segundo algum critério contém todos os elementos possíveis a serem adicionados na solução. Em seguida, emprega-se o valor α para determinar uma lista de candidatos restrita (LCR) contendo os melhores candidatos de LC . Sendo $|LC|$ e $|LCR|$ o tamanho ou quantidade de elementos de LC e LCR respectivamente, $|LCR|$ será o máximo valor entre 1 e $\alpha \times |LC|$, isto é,

Equação 6

$$|LCR| = \text{MAX}(1, \alpha \times |LC|)$$

Por fim, um candidato é escolhido aleatoriamente de LCR e é adicionado à solução em construção. Repete-se o processo até que uma solução completa seja obtida. Melhora-se esta solução durante a segunda fase aplicando movimentos que permitem a exploração das soluções vizinhas àquela. Armazena-se a solução resultante caso esta seja a mais superior até aquela iteração. Repete-se o procedimento até que o critério de parada tenha sido satisfeito. Este critério pode ser um número máximo de iterações, tempo de execução ou iterações sem encontrar melhores soluções.

O comportamento da metaheurística encontra-se resumidamente na Figura 8: inicializam-se a melhor solução conhecida (S^*) e o seu valor de FO (linhas 1 e 2). Nas linhas 4 e 5 uma nova solução é construída utilizando o parâmetro de aleatoriedade e em seguida, melhorada. A melhor solução conhecida é atualizada ao localizar uma solução superior a esta (linhas 6 a 8). Os procedimentos das linhas 4 a 8 são repetidos, gerando uma nova solução a cada iteração, enquanto o critério de parada não for alcançado.

Nas subseções seguintes serão apresentados a fase de construção de soluções viáveis e o método GRASP Reativo implementados.

GRASP (α)	
1	$S^* \leftarrow \emptyset$
2	$FO(S^*) \leftarrow \infty$
3	enquanto não CritérioParada faça
4	$S_1 \leftarrow \text{Construção_Solução}(\alpha)$
5	$S_2 \leftarrow \text{Busca_Local}(S_1)$
6	se $FO(S_2) < FO(S^*)$ então
7	$S^* \leftarrow S_2$
8	fim-se
9	fim-enquanto
10	retorne S^*
fim;	

Figura 8 - Algoritmo GRASP. Adaptado de (FEO; RESENDE, 1995).

4.2.1 Construção de soluções viáveis

Na fase de construção, uma solução viável (conjunto de rotas) é criada a partir da aplicação da Função de Ranqueamento de Nó (*NRF*) (Equação 1) proposto em (GOMES *et al.*, 2010). Um elemento viável é todo vértice que ainda possua solicitação de embarque e/ou desembarque e que pode ser adicionado à rota satisfazendo as restrições de capacidade do veículo e janela de tempo de embarque. A cada iteração do algoritmo Construção_Solução mostrado na Figura 9, uma lista de candidatos (*LC*) de elementos viáveis da rota corrente é preparada, ordenada decrescentemente a partir do valor *NRF* atribuído para cada elemento. Quanto maior for o valor *NRF* atribuído, mais interessante torna-se a utilização do elemento pois o cálculo baseia-se na ponderação entre as perspectivas do veículo e passageiro. Em seguida, uma certa quantidade de primeiros melhores elementos são selecionados para compor a lista de candidatos restritos (*LCR*). O tamanho ou quantidade de elementos desta lista ($|LCR|$) é determinado pela Equação 6 utilizando o parâmetro de entrada α . Em seguida o algoritmo seleciona aleatoriamente um elemento de *LCR* e insere na rota em construção. Note que caso o parâmetro α assuma o valor zero, a escolha do próximo elemento ocorrerá aplicando o critério guloso, pois *LCR* conterá apenas o melhor elemento de *LC*. Caso o parâmetro α assuma valor 1, a escolha será completamente aleatória, pois *LCR* conterá todos os elementos de *LC* e assim qualquer um será selecionado. Ocorrendo a situação de uma rota não puder atender

nenhuma solicitação de transporte restante, esta será finalizada. Havendo alguma solicitação ainda não atendida e outros veículos disponíveis, uma nova rota para outro veículo é iniciada e construída da mesma forma. Repete-se o processo até que não haja veículos disponíveis ou solicitações de transportes viáveis que possam ser satisfeitas. As rotas encontradas resultarão na solução final e seu custo será calculado, sendo que esta solução será utilizada na fase de busca local.

Na Figura 9 encontra-se o algoritmo desta fase. A solução que será construída (S) e o conjunto de solicitações que não foram atendidas (U) são inicializados (linhas 1 e 2). O comportamento para construção de rotas descrito nas linhas 4 a 22 é executado enquanto existirem solicitações a serem atendidas e veículos disponíveis (linha 3). Nas linha 4 e 5 são inicializados a rota atual (R) e sua variável de status (*corrente_nó*), além de retirar um veículo para uso nesta rota. Um nó é selecionado e acrescentado à rota enquanto houver solicitações a serem atendidas (linha 6). Nas linhas 7 a 10 ocorrem a construção das listas com nós candidatos e a escolha aleatória do próximo item (vértice) a ser utilizado. Em seguida, são atualizados os dados de solicitações, a rota e seu status (linhas 11 a 13). Quando não houver mais solicitações, a rota é finalizada e adicionada ao conjunto solução S (linhas 15 a 18). Havendo solicitações no conjunto U , o conjunto P é atualizado (linhas 19 a 22).

Construção_Solução(α)	
1	$S \leftarrow \emptyset$
2	$U \leftarrow \emptyset$
3	enquanto $P \neq \emptyset \wedge K \neq \emptyset$ faça
4	inicialize a rota com o depósito: $R \leftarrow \{0\}$; $corrente_nó \leftarrow 0$
5	$K \leftarrow K - \{\text{veículo escolhido para rota}\}$
6	enquanto $P \neq \emptyset$ faça
7	construa Lista de Candidatos (LC) do $corrente_nó$ com elementos viáveis
8	ordene decrescentemente LC por NRF
9	construa Lista de Candidatos Restritos (LCR) usando o corrente α : $LCR \leftarrow$ primeiros $\text{MAX}(1, \alpha \times LC)$ elementos de LC
10	selecione aleatoriamente o próximo nó a partir da LCR : $próximo_nó \leftarrow LCR[\text{Random}(1, LCR)]$
11	atualize os dados de solicitações, possivelmente removendo as solicitações já satisfeitas e movendo as inviáveis para conjunto U
12	adicione o próximo nó à rota: $R \leftarrow R + próximo_nó$
13	“mova-se” para esta posição: $corrente_nó \leftarrow próximo_nó$
14	fim-enquanto
15	se $P = \emptyset$ então
16	feche a rota R adicionando o depósito (0) no fim
17	adicione R ao conjunto solução S , $S \leftarrow S \cup R$
18	fim-se
19	se $U \neq \emptyset$ então
20	$P \leftarrow U$
21	$U \leftarrow \emptyset$
22	fim-se
23	fim-enquanto
24	retorne S
fim;	

Figura 9 - Algoritmo da fase de construção do GRASP Reativo. Adaptado de (GOMES *et al.*, 2010).

4.2.2 Método GRASP Reativo

O método GRASP Reativo, proposto em (PRAIS; RIBEIRO, 2000), consiste em escolher o parâmetro α , utilizado na fase construtiva, a partir de um conjunto A de v valores previamente fixados para este parâmetro, considerando a qualidade das soluções obtidas anteriormente, isto é, valores do parâmetro que resultarem em melhores soluções terão maior probabilidade de serem escolhidos. Este processo é realizado da seguinte forma: armazenam-se v valores possíveis para o parâmetro α no conjunto A . Para cada um destes valores, são inicializados seu contador de ocorrências ($count[i]$) e somatório da função objetivo (FO) ($score[i]$) com zero e sua probabilidade de escolha (P_i) de maneira equiprovável ($1/v$). A cada iteração do GRASP Reativo, inicialmente um parâmetro α é escolhido probabilisticamente do conjunto A para a fase construtiva e no fim são contabilizados a ocorrência do parâmetro e a qualidade de sua FO, assim como ocorre o auto-ajuste do fator estatístico da seleção a cada γ iterações. O ajuste da probabilidade associado a cada elemento de A faz uso de $count$, $score$, melhor valor de FO conhecido ($FO(S^*)$) e um parâmetro de amplificação θ para ampliar a diferença entre os valores Q_i , que são empregados diretamente no recálculo da probabilidade de escolha P_i .

O mecanismo reativo foi implementado segundo o apresentado no trabalho (KAMPKE, 2010). Na Figura 10 encontra-se o algoritmo do GRASP Reativo. Nas linhas 1 a 9 são inicializados a melhor solução conhecida (S^*) e seu valor FO, o contador de iterações ($num_iterations$), o conjunto de valores pré-fixados para o parâmetro α (A) e variáveis associadas a este conjunto: contador de ocorrências ($count[i]$), somatório da função objetivo (FO) ($score[i]$) e probabilidade de escolha (P_i). Uma solução é construída usando um parâmetro selecionado probabilisticamente e em seguida, melhorada (linhas 11 a 13). Caso o resultado seja o melhor conhecido até aquela iteração, a melhor solução é atualizada (linhas 14 a 16). Depois é contabilizado a ocorrência do parâmetro α utilizado e a qualidade de sua FO (linhas 17 e 18), assim como a cada γ iterações é realizado o processo de ajuste do fator estatístico da seleção (linhas 19 a 24). Quanto maior a média da solução ($avg[i]$), menor será Q_i e, conseqüentemente, a P_i recalculada. Finalmente é atualizado o contador de iterações (linha 25). Repete-se o procedimento das linhas 11 a 25 enquanto não for alcançado o critério de parada ($MAX_ITERATIONS$ executados).

GRASP_Reativo (*MAX_ITERATIONS*, *tipo_estrategia*, *tipo_movimento*)

```
1       $S^* \leftarrow \emptyset$ 
2       $FO(S^*) \leftarrow \infty$ 
3       $num\_iterations \leftarrow 1$ 
4      Defina um conjunto de possíveis  $\alpha$ :  $A \leftarrow \{\alpha_1, \alpha_2, \dots, \alpha_v\}$ 
5      para  $i \leftarrow 1$  até  $v$  faça:
6           $count[i] \leftarrow 0$ 
7           $score[i] \leftarrow 0$ 
8           $P_i \leftarrow 1/v$ 
9      fim-para
10     enquanto  $num\_iterations \leq MAX\_ITERATIONS$  faça
11          $\alpha \leftarrow$  selecione  $\alpha_i \in A$ ,  $i \in \{1, 2, \dots, v\}$ , com probabilidade de
            escolha  $P_i$ 
12          $S_1 \leftarrow$  Construção_Solução( $\alpha$ )
13          $S_2 \leftarrow$  Busca_Local( $S_1$ , tipo_estrategia, tipo_movimento)
14         se  $FO(S_2) < FO(S^*)$  então
15              $S^* \leftarrow S_2$ 
16         fim-se
17          $count[i] \leftarrow count[i] + 1$ 
18          $score[i] \leftarrow score[i] + FO(S_2)$ 
19         se  $num\_iterations \bmod \gamma = 0$  então
20              $avg[i] \leftarrow score[i] / count[i] \quad \forall i \in \{1, 2, \dots, v\}$ 
21              $Q_i \leftarrow (FO(S^*) / avg[i])^\theta \quad \forall i \in \{1, 2, \dots, v\}$ 
22              $\sigma \leftarrow \sum Q_i$ 
23              $P_i \leftarrow Q_i / \sigma \quad \forall i \in \{1, 2, \dots, v\}$ 
24         fim-se
25          $num\_iterations \leftarrow num\_iterations + 1$ 
26     fim-enquanto
27     retorne  $S^*$ 
```

fim;

Figura 10 - Algoritmo GRASP_Reativo. Adaptado de (KAMPKE, 2010).

4.3 Busca Local

A Busca Local é um procedimento adotado quando deseja-se melhorar a qualidade da solução encontrada através de pequenas modificações (movimentos) na estrutura desta, permitindo a exploração de sua vizinhança. Vizinhança de uma solução é o conjunto de soluções obtidas por meio de movimentos sobre aquela solução. O procedimento inicia-se com uma solução inicial, e a cada etapa de busca, uma melhor solução encontrada na vizinhança substitui a solução atual. O procedimento termina quando atinge um mínimo local, ou seja, quando não há mais melhora na qualidade da solução corrente (ela é melhor que todas as soluções de sua vizinhança).

Para a fase de Busca Local foram desenvolvidos e utilizados os seguintes tipos de movimento: Troca, Substituição, Deslocamento e Aleatório. O movimento é aplicado sobre um vetor que é resultado da concatenação das rotas de uma dada solução, retirando-se o depósito. Desta forma, os movimentos podem ser entre elementos de uma mesma rota ou rotas diferentes. A Figura 11 ilustra uma solução composta de duas rotas.

Rota 1:	Depósito	1	9	1	Depósito	
Rota 2:	Depósito	2	7	8	3	Depósito

Figura 11 - Representação de uma solução com duas rotas.

A solução da Figura 11 será representada pelo seguinte vetor de sete posições, que é a concatenação das rotas, sem o depósito (Figura 12):

0	1	2	3	4	5	6
1	9	1	2	7	8	3

Figura 12 - Representação concatenada de uma solução com duas rotas.

Considerando o vetor exibido na Figura 12 como solução atual ou corrente, os movimentos foram definidos da seguinte maneira:

- Troca: consiste na escolha de duas posições deste vetor com conteúdo diferente e realiza a troca dos valores contidos nos mesmos. No exemplo da Figura 13 foram escolhidas as posições 1 e 3, e assim os nós 2 e 9 são trocados de posições.

0	<u>1</u>	2	<u>3</u>	4	5	6
1	2	1	9	7	8	3

Figura 13 - Exemplo da aplicação do movimento Troca.

- Substituição: consiste na escolha de uma posição deste vetor e a substituição do conteúdo deste por outro, contido no conjunto de vértices de parada $V-\{0\}$. Neste exemplo (Figura 14), a posição 1 foi escolhida e o nó 9 foi substituído por 10.

0	<u>1</u>	2	3	4	5	6
1	10	1	2	7	8	3

Figura 14 - Exemplo da aplicação do movimento Substituição.

- Deslocamento: consiste na escolha de duas posições distintas deste vetor, remoção do conteúdo da primeira posição, realização de um deslocamento à esquerda do elemento contido na segunda posição escolhida e quaisquer elementos entre essas posições (ou deslocamento à direita, dependendo de sua posição relativa à primeira posição escolhida) e inserção do elemento removido da primeira posição na segunda. Como exemplo de deslocamento à esquerda (Figura 15), as posições 0 e 3 do vetor atual foram escolhidas nesta ordem. Em seguida, remove-se o nó 1, deslocam-se à esquerda os elementos contidos nas posições anteriores ou igual a 3 e em seguida, insere-se o nó 1 nesta posição.

<u>0</u>	1	2	<u>3</u>	4	5	6
9	1	2	1	7	8	3

Figura 15 - Exemplo da aplicação do movimento Deslocamento à esquerda.

Para ilustrar o deslocamento à direita (Figura 16), as posições 3 e 0 do vetor atual foram escolhidas, nesta ordem. Em seguida, remove-se o nó 2, deslocam-se à direita os elementos contidos nas posições posteriores ou igual a 0 e em seguida, insere-se o nó 2 nesta posição.

<u>0</u>	1	2	<u>3</u>	4	5	6
2	1	9	1	7	8	3

Figura 16 - Exemplo da aplicação do movimento Deslocamento à direita.

- Aleatório: consiste na geração de um vizinho escolhendo aleatoriamente um dos movimentos citados anteriormente.

Para cada movimento foram adotadas duas estratégias possíveis de escolha da próxima solução corrente: Melhor Vizinho e Primeiro Melhor Vizinho.

- Melhor Vizinho: consiste na exploração de toda vizinhança de uma solução fornecida utilizando um movimento pré-determinado e retorna a solução da vizinhança com melhor valor de FO ou a própria solução fornecida, caso não haja outra melhor na vizinhança.
- Primeiro Melhor Vizinho: consiste na exploração iterativa da vizinhança de uma solução fornecida utilizando um movimento pré-determinado apenas até encontrar a primeira solução com melhor valor de FO na vizinhança. Caso não seja encontrada, a própria solução fornecida é retornada.

Considerando o tamanho da rota concatenada $|R|$, o movimento Troca e Deslocamento geram um máximo de $|R| \times (|R| - 1)/2$ e $(|R| - 1)^2$ vizinhos, respectivamente. Sendo este total atingido apenas quando todos os elementos são distintos entre si, uma vez que estes movimentos não são efetuados para elementos iguais.

Considerando a quantidade de vértices de parada $|V|$, o movimento Substituição gera um máximo de $|R| \times (|V| - 1)$ vizinhos.

Como o movimento Aleatório não é determinístico como os demais, foi determinado que a quantidade máxima de vizinhos gerada seria de $0.1 \times |R|^2$. Este total é atingido somente se todos os vizinhos forem distintos entre si, já que o movimento não utiliza memória para armazenar os vizinhos gerados.

Na Figura 17 é apresentado o algoritmo de Busca Local. Ele foi implementado para suportar os dois tipos de estratégias a serem utilizados: Primeiro Melhor e Melhor Vizinho. O algoritmo comporta-se da seguinte maneira: uma vez determinado qual o tipo de estratégia será utilizado, a solução atual é armazenada e busca-se o vizinho daquela solução que satisfaça a estratégia escolhida. O processo é repetido até que não seja mais possível melhorar a solução, isto é, a estratégia não encontrou um vizinho melhor na vizinhança (a solução retornada pela estratégia é igual a solução atual).

Busca_Local (S , $tipo_estrategia$, $tipo_movimento$)	
1	$S_1 \leftarrow S$
2	se $tipo_estrategia = MelhorVizinho$ então
3	faça
4	$S_2 \leftarrow S_1$
5	$S_1 \leftarrow MelhorVizinho(S_2, tipo_movimento)$
6	enquanto $S_1 \neq S_2$
7	fim-se
8	senão
9	faça
10	$S_2 \leftarrow S_1$
11	$S_1 \leftarrow PrimeiroMelhorVizinho(S_2, tipo_movimento)$
12	enquanto $S_1 \neq S_2$
13	fim-senão
14	retorne S_2
fim;	

Figura 17 - Busca Local com tipos de estratégias diferentes.

Na Figura 18 é apresentado o algoritmo da estratégia Melhor Vizinho. O comportamento desta estratégia é o seguinte: inicialize a solução S^* com a solução fornecida. Em seguida, para cada vizinho gerado a partir da solução fornecida utilizando o movimento determinado, verifique se este vizinho possui valor de FO melhor que o de S^* . Em caso afirmativo, atualize a solução S^* utilizando este vizinho. Ao final do método é retornado o melhor vizinho encontrado ou na ausência dele, a própria solução fornecida.

Na Figura 19 é apresentado o algoritmo da estratégia Primeiro Melhor Vizinho. O algoritmo funciona da seguinte forma: a solução fornecida é armazenada na solução S^* . Em seguida, no momento em que for encontrado um vizinho, gerado a partir da solução fornecida utilizando o movimento determinado, que possua melhor valor de FO que o de S^* , este vizinho é armazenado em S^* e retornado, interrompendo o processo de busca. Caso este não seja encontrado, a própria solução fornecida é retornada.

MelhorVizinho ($S, tipo_movimento$)

```
1       $S^* \leftarrow S$ 
2      para cada  $Vizinho$  gerado a partir de  $S$  utilizando  $tipo\_movimento$ 
3          se  $FO(Vizinho) < FO(S^*)$ 
4               $S^* \leftarrow Vizinho$ 
5          fim-se
6      fim-para
7      retorne  $S^*$ 
```

fim;

Figura 18 - Estratégia Melhor Vizinho com tipo de movimento genérico.

PrimeiroMelhorVizinho ($S, tipo_movimento$)

```
1       $S^* \leftarrow S$ 
2      para cada  $Vizinho$  gerado a partir de  $S$  utilizando  $tipo\_movimento$ 
3          se  $FO(Vizinho) < FO(S^*)$ 
4               $S^* \leftarrow Vizinho$ 
5              retorne  $S^*$ 
6          fim-se
7      fim-para
8      retorne  $S^*$ 
```

fim;

Figura 19 - Estratégia Primeiro Melhor Vizinho com tipo de movimento genérico.

4.4 GRASP Reativo + Busca Tabu

Com a finalidade de evitar mínimos locais e obter melhora dos resultados encontrados, foi utilizado o algoritmo GRASP Reativo (Figura 10) substituindo-se a Busca Local da linha 13 por uma Busca Tabu, uma vez que é um procedimento capaz de guiar e melhorar o processo de exploração de vizinhança.

A Busca Tabu (BT), proposta em (GLOVER, 1989), é um método de Busca Local que permite a piora na qualidade da solução corrente a fim de escapar de ótimos locais. Isso é possível através da utilização de uma lista (lista tabu) que armazena certos movimentos já realizados como proibidos (tabu), por um determinado número de iterações. Durante o processo de exploração, a solução atual

somente é atualizada pelo melhor vizinho desta caso o movimento que gerou a melhoria não seja proibido (tabu) ou então, atenda a um critério de aspiração, como por exemplo, a nova solução seja melhor que todas as soluções conhecidas. Em seguida, o movimento inverso ao que gerou esta melhora é acrescentada à lista tabu, impedindo assim que tal movimento seja desfeito durante um certo número de iterações.

Na Figura 20 encontra-se o algoritmo da Busca Tabu utilizado. Da linha 1 à 4 são inicializados: a melhor solução conhecida (S^*) e a solução corrente (S_I) com a solução recebida S e os contadores da iteração corrente (*iteração*) e a da melhor solução (*melhorIteração*) com 0. Na linha 6 é atualizada a quantidade de iterações decorridas até o momento. Na linha 7 ocorre a busca pelo melhor vizinho da solução corrente que não foi gerado por um movimento tabu ou que satisfaz o critério de aspiração. Na linha 8 o movimento inverso ao que gerou este melhor vizinho é armazenado na lista por um número de iterações (*duraçãoTabu*). Depois atualiza-se a solução corrente com a solução encontrada (linha 9). Em seguida, caso esta solução seja melhor que todas as conhecidas, a solução S^* é atualizada e *melhorIteração* é atualizada com a iteração corrente (linhas 10 a 13). O processo da linha 6 à 13 repete-se até que tenham ocorrido *BTMAX* iterações sem melhora da solução.

Visto que os melhores resultados foram obtidos em sua maioria ao utilizar o movimento de Substituição com estratégia de Melhor Vizinho, conforme veremos a seguir, esta configuração foi a escolhida e implementada como estratégia de exploração de vizinhança na Busca Tabu. Considerando o conjunto dos vértices de parada exceto depósito ($V - \{0\}$) e o tamanho da rota concatenada $|R|$, a lista tabu é representada pela matriz *listaTabu*[*nó*][*posição*], onde *posição* $\in \{0, 1, \dots, |R| - 1\}$ e *nó* $\in V - \{0\}$.

Busca_Tabu($S, BTMAX$)	
1	$S^* \leftarrow S$
2	$S_1 \leftarrow S$
3	$iteração \leftarrow 0$
4	$melhorIteração \leftarrow 0$
5	enquanto $iteração - melhorIteração \leq BTMAX$
6	$iteração \leftarrow iteração + 1$
7	$S_2 \leftarrow$ melhor vizinho de S_1 em que o movimento de substituição M que gerou este vizinho não é tabu ($ListaTabu[M] < iteração$) ou, se tabu, satisfaça o critério de aspiração ($FO(vizinho) < FO(S^*)$)
8	$ListaTabu[M] \leftarrow iteração + duraçãoTabu$
9	$S_1 \leftarrow S_2$
10	se $FO(S_2) < FO(S^*)$ então
11	$S^* \leftarrow S_2$
12	$melhorIteração \leftarrow iteração$
13	fim-se
14	fim-enquanto
15	retorne S^*
fim;	

Figura 20 - Algoritmo da Busca Tabu.

Segue-se um exemplo para demonstrar o funcionamento da lista tabu utilizando-se o movimento Substituição. Para fins didáticos, na Figura 21, é apresentado a representação concatenada de uma solução semelhante ao apresentado na Figura 12:

0	1	2	3	4	5	6
3	9	3	5	7	8	1

Figura 21 - Exemplo de rota concatenada semelhante ao apresentado na Figura 12.

Considere no algoritmo Busca Tabu, $BTMAX = 10$ e $duraçãoTabu = 5$. Suponha que na iteração 1, a posição 2 foi escolhida e o nó 3 foi substituído por 4. Logo será acrescentada na lista tabu a seguinte proibição: $listaTabu[4][2] = iteração + duraçãoTabu$. Isto significa que a posição 2 será impedida de desfazer o movimento (retirar o elemento 4 da posição 2) até a 6ª iteração, ou seja, durante 5

iterrações. A Figura 22 contém o vetor resultante e a Figura 23 a lista tabu com a proibição do movimento realizado ser desfeito.

0	1	<u>2</u>	3	4	5	6
3	9	4	5	7	8	1

Figura 22 - Exemplo da aplicação do movimento Substituição dentro da Busca Tabu.

	0	1	2	3	4	5	6
1							
2							
3							
4			6				
5							
6							
7							
8							
9							

Figura 23 - Exemplo de Lista Tabu iteração 1.

Portanto, a partir da iteração 7 o movimento deixa de estar proibido uma vez que $\text{listaTabu}[4][2] \neq 7$.

4.5 Replanejamento

No caso dinâmico do PRVDTRD, solicitações aparecem em tempo real durante a execução das rotas. Para o tratamento adequado desta situação (veículos em trânsito, passageiros à bordo e outros já confirmados) e evitar e/ou reduzir o desperdício computacional na descoberta de novas soluções, foram propostos algoritmos de replanejamento com critério guloso (RepGuloso) e parâmetro aleatório auto-ajustável (RepGR). Estes procuram viabilizar a solução em execução considerando o status do veículo num certo horário e a inserção das novas solicitações. Logo, são necessários a existência de uma solução em execução (*Sinicial*), um horário para determinar e atualizar o status do veículo (trecho da rota percorrido ou em execução, solicitações atendidas ou em atendimento - passageiro à bordo) e um conjunto contendo as novas solicitações que serão acrescentadas ao processo de atendimento (*conjuntoNovasSolicitações*).

O replanejamento RepGuloso (Figura 24) emprega a função NRF (Equação 1) no processo de reconstrução das suas rotas, de maneira que, segundo esta função, é selecionado o melhor nó a ser inserido a cada iteração. Uma vez que o algoritmo assemelha-se ao Construção_Solução(α) (Figura 9), as diferenças são comentadas a seguir. As solicitações não atendidas pela solução *Sinicial* e as novas solicitações (*conjuntoNovasSolicitações*) são adicionadas ao conjunto de solicitações a serem atendidas (P) (linha 3). O comportamento das linhas 5 a 23 repete-se para cada rota R da solução inicial. Nas linhas de 5 a 8, é atualizado o status da execução da rota. Nas linhas 10 e 11, ocorre o processo de reconstrução da rota.

Já o replanejamento RepGR (Figura 25) tem um parâmetro de aleatoriedade associado ao processo de replanejamento de suas rotas e utiliza-se de mecanismo reativo para autocalibrar as probabilidades associadas a escolha deste parâmetro. O mecanismo reativo assemelha-se ao descrito para o GRASP Reativo. De maneira análoga realizada para o RepGuloso, somente as diferenças entre RepGR e GRASP Reativo (Figura 10) são comentadas. Na linha 14, as solicitações não atendidas pela solução *Sinicial* e as novas solicitações são adicionadas ao conjunto P . Nas linhas de 16 a 19, é atualizado o status da execução da rota. Nas linhas de 21 a 24, ocorre o processo de reconstrução da rota utilizando o parâmetro de aleatoriedade. O comportamento das linhas 16 a 36 repete-se para cada rota R da solução inicial. Cabe ressaltar que nenhum método de melhoria é empregado sobre a solução reconstruída, diferentemente do GRASP Reativo, que utiliza o método Busca Local.

RepGuloso (*Sinicial*, *horarioT*, *conjuntoNovasSolicitações*)

```
1       $S \leftarrow \emptyset$ 
2       $U \leftarrow \emptyset$ 
3       $P \leftarrow \text{conjuntoNovasSolicitações} \cup \text{solicitações não atendidas em Sinicial}$ 
4      para cada rota  $R$  na solução Sinicial faça
5          remova de  $R$  os vértices de parada que serão visitados após horarioT
6          atualize a posição atual do veículo:  $\text{corrente\_nó} \leftarrow$  último nó da rota
               $R$ 
7          atualize o status da lotação do veículo
8          adicione em  $P$  as solicitações dos passageiros que serão atendidos ou
              desembarcarão após horarioT
9          enquanto  $P \neq \emptyset$  faça
10              a partir do  $\text{corrente\_nó}$ , construa  $NRF$  apenas com nós viáveis
11              selecione o próximo nó a partir do maior valor atribuído de
                   $NRF$ :  $\text{próximo\_nó} \leftarrow \arg( \text{MAX}( NRF[i] ) )$ 
12              atualize os dados de solicitações, possivelmente removendo as
                  solicitações já satisfeitas e movendo as inviáveis para
                  conjunto  $U$ 
13              adicione o próximo nó à rota:  $R \leftarrow R + \text{próximo\_nó}$ 
14              “mova-se” para esta posição:  $\text{corrente\_nó} \leftarrow \text{próximo\_nó}$ 
15          fim-enquanto
16          se  $P = \emptyset$  então
17              feche a rota  $R$  adicionando o depósito (0) no fim
18              adicione  $R$  ao conjunto solução  $S$ ,  $S \leftarrow S \cup R$ 
19          fim-se
20          se  $U \neq \emptyset$  então
21               $P \leftarrow U$ 
22               $U \leftarrow \emptyset$ 
23          fim-se
24      fim-para
25      retorne  $S$ 
```

fim;

Figura 24 - Algoritmo do replanejamento RepGuloso.

RepGR (*MAX_ITERATIONS*, *Sinicial*, *horarioT*, *conjuntoNovasSolicitações*)

```
1       $S^* \leftarrow \emptyset$ 
2       $FO(S^*) \leftarrow \infty$ 
3       $num\_iterations \leftarrow 1$ 
4      Defina um conjunto de possíveis  $\alpha$ :  $A \leftarrow \{\alpha_1, \alpha_2, \dots, \alpha_v\}$ 
5      para  $i \leftarrow 1$  até  $v$  faça:
6           $count[i] \leftarrow 0$ 
7           $score[i] \leftarrow 0$ 
8           $P_i \leftarrow 1/v$ 
9      fim-para
10     enquanto  $num\_iterations \leq MAX\_ITERATIONS$  faça
11          $\alpha \leftarrow$  selecione  $\alpha_i \in A$ ,  $i \in \{1, 2, \dots, v\}$ , com probabilidade de escolha
             $P_i$ 
12          $S \leftarrow \emptyset$ 
13          $U \leftarrow \emptyset$ 
14          $P \leftarrow conjuntoNovasSolicitações \cup$  solicitações não atendidas em
            Sinicial
15         para cada rota  $R$  na solução Sinicial faça
16             remova de  $R$  os vértices de parada que serão visitados após
                horarioT
17             atualize a posição atual do veículo:  $corrente\_nó \leftarrow$  último nó
                da rota  $R$ 
18             atualize o status da lotação do veículo
19             adicione em  $P$  as solicitações dos passageiros que serão
                atendidos ou desembarcarão após horarioT
20         enquanto  $P \neq \emptyset$  faça
21             construa Lista de Candidatos ( $LC$ ) do  $corrente\_nó$  com
                elementos viáveis
22             ordene decrescentemente  $LC$  por  $NRF$ 
23             construa Lista de Candidatos Restritos ( $LCR$ ) usando o
                corrente  $\alpha$ :  $LCR \leftarrow$  primeiros  $MAX(1, \alpha \times |LC|)$ 
                elementos de  $LC$ 
24             selecione aleatoriamente o próximo nó a partir da
```

```

                 $LCR: próximo\_nó \leftarrow LCR[Random(1, |LCR|)]$ 
25         atualize os dados de solicitações, possivelmente
                removendo as solicitações já satisfeitas e movendo
                as inviáveis para conjunto  $U$ 
26         adicione o próximo nó à rota:  $R \leftarrow R + próximo\_nó$ 
27         “mova-se” para esta posição:  $corrente\_nó \leftarrow$ 
                 $próximo\_nó$ 
28         fim-enquanto
29         se  $P = \emptyset$  então
30             feche a rota  $R$  adicionando o depósito (0) no fim
31             adicione  $R$  ao conjunto solução  $S$ ,  $S \leftarrow S \cup R$ 
32         fim-se
33         se  $U \neq \emptyset$  então
34              $P \leftarrow U$ 
35              $U \leftarrow \emptyset$ 
36         fim-se
37         fim-para
38         se  $FO(S) < FO(S^*)$  então
39              $S^* \leftarrow S$ 
40         fim-se
41          $count[i] \leftarrow count[i] + 1$ 
42          $score[i] \leftarrow score[i] + FO(S)$ 
43         se  $num\_iterations \bmod \gamma = 0$  então
44              $avg[i] \leftarrow score[i] / count[i] \quad \forall i \in \{1, 2, \dots, v\}$ 
45              $Q_i \leftarrow (FO(S^*) / avg[i])^\theta \quad \forall i \in \{1, 2, \dots, v\}$ 
46              $\sigma \leftarrow \sum Q_i$ 
47              $P_i \leftarrow Q_i / \sigma \quad \forall i \in \{1, 2, \dots, v\}$ 
48         fim-se
49          $num\_iterations \leftarrow num\_iterations + 1$ 
50     fim-enquanto
51     retorne  $S^*$ 

```

fim;

Figura 25 - Algoritmo do replanejamento RepGR.

5 EXPERIMENTAÇÃO COMPUTACIONAL

Neste capítulo são apresentados os resultados da aplicação dos métodos propostos para o PRVDTRD a um conjunto de instâncias com dados gerados aleatoriamente. Também são expostos os parâmetros dos algoritmos e justificativas de sua utilização.

O capítulo foi organizado da seguinte forma: a seção 5.1 contém informações relevantes sobre o ambiente e conjunto de dados utilizados para os testes; a seção 5.2 contém os parâmetros necessários para o uso dos métodos; e na seção 5.3 são expostos os resultados e análise de desempenho entre os métodos propostos e os implementados segundo a literatura.

5.1 Ambiente de testes e instâncias utilizadas

Para execução dos testes foi utilizado um computador que possui processador Intel Core i5-2410M com 2,30GHz e 4 GB de RAM, sem o uso de multiprocessamento. Os algoritmos dos métodos disponíveis na literatura e os propostos neste trabalho foram implementados na linguagem de programação Java, na versão 1.6.0.260. Os testes foram executados usando o compilador JDK 6.0.

Uma vez que não foram encontradas instâncias disponíveis na literatura para o PRVDTRD foi adotado o mesmo procedimento que em (GOMES *et al.*, 2010): geração aleatória das próprias instâncias. Cada instância possui as seguintes informações:

- Quantidade de solicitações presente no conjunto $P = 110$;
- Quantidade de veículos disponíveis na frota $K \in \{4, 5, 6\}$;
- Capacidade Q de cada veículo $\in \{10, 15\}$;
- Solicitações: possuem local de embarque, desembarque e janelas de tempo gerados aleatoriamente, sendo estas últimas pertencentes a um horizonte de três horas.

As solicitações geradas e utilizadas estão disponíveis no APÊNDICE A.

Foram geradas 6 instâncias e a notação utilizada para nomeá-las é $Ps-Kv-Qc$, onde s é o número de solicitações, v é a quantidade de veículos disponíveis e c , a capacidade de cada veículo.

Tendo em vista que são algoritmos heurísticos com componentes aleatórios, para melhor comparação entre resultados dos métodos, todos foram executados 10 vezes para cada instância. A partir daí, para cada método e para cada instância, foram identificados o melhor valor de FO obtido (Melhor FO) e a média do valor das 10 execuções (FO Médio).

5.2 Parâmetros dos métodos propostos e sua utilização nos testes

Para utilização dos métodos heurísticos disponíveis na literatura e/ou propostos neste trabalho, são necessários alguns parâmetros que definem a calibragem com melhor desempenho para o problema. Em alguns casos, os parâmetros foram definidos segundo os utilizados na literatura.

5.2.1 Parâmetros das funções de avaliação

Os pesos atribuídos aos critérios distância (α_d), número de passageiros (α_p), janela de tempo de desembarque (α_v) e janela de tempo de embarque (α_t), utilizados para cálculo da função NRf , foram os mesmos apresentados no trabalho de (GOMES *et al.*, 2010): $\alpha_d = 0.20$; $\alpha_p = 0.15$; $\alpha_v = 0.55$; $\alpha_t = 0.10$.

A função objetivo (FO) apresentada no capítulo 4 (Equação 5) busca minimizar a quantidade de veículos utilizados e a distância total percorrida por eles, além da quantidade de solicitações que não foram atendidas. Para cálculo desta função foram definidos o custo fixo para utilização do veículo (W) e pelo não atendimento de uma solicitação (Y) como $W = 1000$ e $Y = 800$. Estes valores foram determinados considerando testes preliminares, tendo em vista que são relativamente altos comparados aos valores de distância percorridos pelos veículos. Desta forma, cada veículo tem custo operacional de 1000 e cada solicitação não atendida 800. Assim, a utilização de um novo veículo torna-se viável quando ele puder atender a pelo menos duas solicitações (deve-se levar em conta ainda o custo de viagem para estes atendimentos).

5.2.2 Parâmetros do GRASP Reativo

O critério de parada utilizado no GRASP Reativo foi o número máximo de iterações ($MAX_ITERATIONS$). Ele foi definido como $MAX_ITERATIONS = 600$.

Além disso, o conjunto A de possíveis α_i que serão utilizados pelo parâmetro de aleatoriedade foi definido como $A = \{0.30; 0.35; 0.40; 0.45; 0.50; 0.55; 0.60; 0.65; 0.70\}$.

O parâmetro de amplificação (θ) que torna possível detectar as diferenças entre os valores Q_i foi definido como $\theta = 10$, uma vez que, em testes preliminares, era o suficiente para evidenciar tais diferenças. De igual modo, o parâmetro γ , que indica o número de iterações para atualizar a probabilidade P_i dos α_i , foi definido como $\gamma = 20$.

5.2.3 Parâmetros do GRASP Reativo + Busca Tabu

Para utilização do GRASP Reativo + Busca Tabu, foram utilizados além dos parâmetros definidos na subseção anterior, o critério de parada da Busca Tabu (número máximo de iterações sem melhoria) e duração tabu utilizado na lista de movimentos proibidos (lista tabu). Estes foram definidos como $BTMAX = 10$ e $duraçãoTabu = 5$ pois foi observado em testes preliminares que outros valores acrescentavam muito pouco à descoberta de novas soluções e este processo demandaria muito tempo.

5.2.4 Parâmetros dos Replanejamentos

Os replanejamentos dependem diretamente do horário especificado para realizar a atualização do status da rota e definir quais passageiros estão a bordo ou ainda não tiveram suas solicitações atendidas.

A fim de comparar o desempenho do replanejamento no caso dinâmico com o encontrado pelos métodos desenvolvidos para o caso estático, era necessário que a quantidade de solicitações para atendimento fosse a mesma. Porém, no caso dinâmico, somente algumas seriam conhecidas previamente e após o horário especificado ($horarioT$) seria feito um replanejamento considerando as já conhecidas que estão em atendimento e as novas que chegam após este horário. Por isso foram definidos os seguintes valores para $horarioT$: 38 e 87. O primeiro valor deixará várias solicitações (40 das 110) serem conhecidas apenas *a posteriori* e o segundo, apenas algumas (mais especificamente, 20 das 110) *a posteriori*.

Os métodos para o caso estático são executados utilizando somente as solicitações conhecidas *a priori*. A solução resultante é utilizada como solução inicial (em execução) pelo algoritmo de replanejamento.

Como o método de replanejamento RepGR utiliza mecanismo reativo e critério de parada semelhante ao GRASP Reativo, foram utilizados os parâmetros definidos na subseção 5.2.2 (Parâmetros do GRASP Reativo).

5.3 Análise e discussão dos resultados dos experimentos computacionais

É importante salientar que para comparação do GRASP Reativo proposto com os métodos apresentados na literatura, os métodos de GOMES *et al.* (2010), heurística Construtiva Gulosa e GRASP-like foram reimplementados seguindo a descrição dada pelos autores, uma vez que não foram disponibilizadas as instâncias utilizadas em seu trabalho. Assim, com a reimplementação, todos os métodos foram executados com as mesmas instâncias, mesma máquina e mesmo estilo e linguagem de programação, possibilitando uma comparação efetiva entre os métodos.

Visto que já foi realizado o detalhamento do ambiente de testes e instâncias utilizadas e a discriminação dos parâmetros utilizados nos testes pelos métodos propostos anteriormente, nesta seção serão apresentados os resultados obtidos na experimentação.

Vale ressaltar que neste trabalho foram desenvolvidos dois métodos para o caso estático (GRASP Reativo e GRASP Reativo + Busca Tabu), dois métodos de replanejamento para o caso dinâmico (RepGuloso e RepGR). Estes métodos utilizam duas estratégias (Primeiro Melhor e Melhor Vizinho) e quatro movimentos (Troca, Substituição, Deslocamento e Aleatório).

Com o objetivo de facilitar o entendimento e demonstrar o bom desempenho dos métodos propostos através da comparação entre si e com os disponíveis na literatura, os resultados e sua respectiva análise e discussão foram agrupados da seguinte maneira: tipos de estratégia e movimento; quantidade e capacidade de veículos; métodos para o caso estático e métodos para o caso dinâmico.

5.3.1 Tipos de Estratégia e Movimento

A fim de determinar a combinação de estratégia e movimento mais adequada para o GRASP Reativo proposto, esta metaheurística foi testada com todas as

combinações de estratégias (Melhor Vizinho e Primeiro Melhor) e movimentos (Troca, Substituição, Deslocamento e Aleatório) apresentados.

Seguem abaixo alguns resultados encontrados.

Na Tabela 3 são apresentados resumidamente, para cada combinação de estratégia e movimento, a média do valor de Melhor FO e FO Médio alcançados e o percentual de instâncias obtido com melhor FO e melhor FO Médio em relação à estratégia utilizada; as informações detalhadas foram disponibilizadas no APÊNDICE B. Conforme mencionado na subseção 5.1, a média dos valores encontrados nas 10 execuções foi denominada FO Médio e o melhor valor encontrado, Melhor FO. Nesta tabela, para cada estratégia, encontra-se em negrito o melhor resultado obtido de Melhor FO e FO Médio. Analisando a estratégia Primeiro Melhor, percebe-se que apesar dos movimentos Substituição, Deslocamento e Aleatório obterem a mesma quantidade de instâncias com melhor FO (33,33%), a Substituição destacou-se em todas as instâncias com melhor FO Médio e sobressaiu-se mais que as outras nos critérios Melhor FO e FO Médio. Já na estratégia Melhor Vizinho, apesar dos movimentos Substituição e Deslocamento terem alcançado a mesma quantidade de instâncias com melhor FO Médio, o movimento Substituição sobressaiu-se mais que os outros no critério FO Médio. Além disso, para cada movimento, ao comparar as estratégias utilizadas, percebe-se que Melhor Vizinho sobrepõe-se ao Primeiro Melhor. A única exceção foi Substituição no critério FO Médio, mas nesse caso a diferença entre as estratégias foi menor que 1%.

Tabela 3 - Média dos valores de Melhor FO e FO Médio e percentual de instâncias com melhor FO e melhor FO Médio obtidas considerando apenas Estratégia e Movimento.

Estratégias	Movimentos	Melhor FO	FO Médio	% de Instâncias com melhor FO	% de Instâncias com melhor FO Médio
Primeiro Melhor	Troca	27785,67	29278,27	0,00%	0,00%
	Substituição	26820,00	28405,97	33,33%	100,00%
	Deslocamento	27183,00	29711,70	33,33%	0,00%
	Aleatório	27340,67	29508,90	33,33%	0,00%
Melhor Vizinho	Troca	26869,00	28929,20	0,00%	16,67%
	Substituição	26497,50	28631,45	33,33%	33,33%
	Deslocamento	25909,33	28871,48	66,67%	33,33%
	Aleatório	27143,83	28991,15	0,00%	16,67%

A Figura 26 ilustra os dados apresentados na Tabela 3 comparando as estratégias segundo os critérios de Melhor FO e FO Médio. Percebe-se que Melhor Vizinho obteve melhores resultados na maioria das vezes.

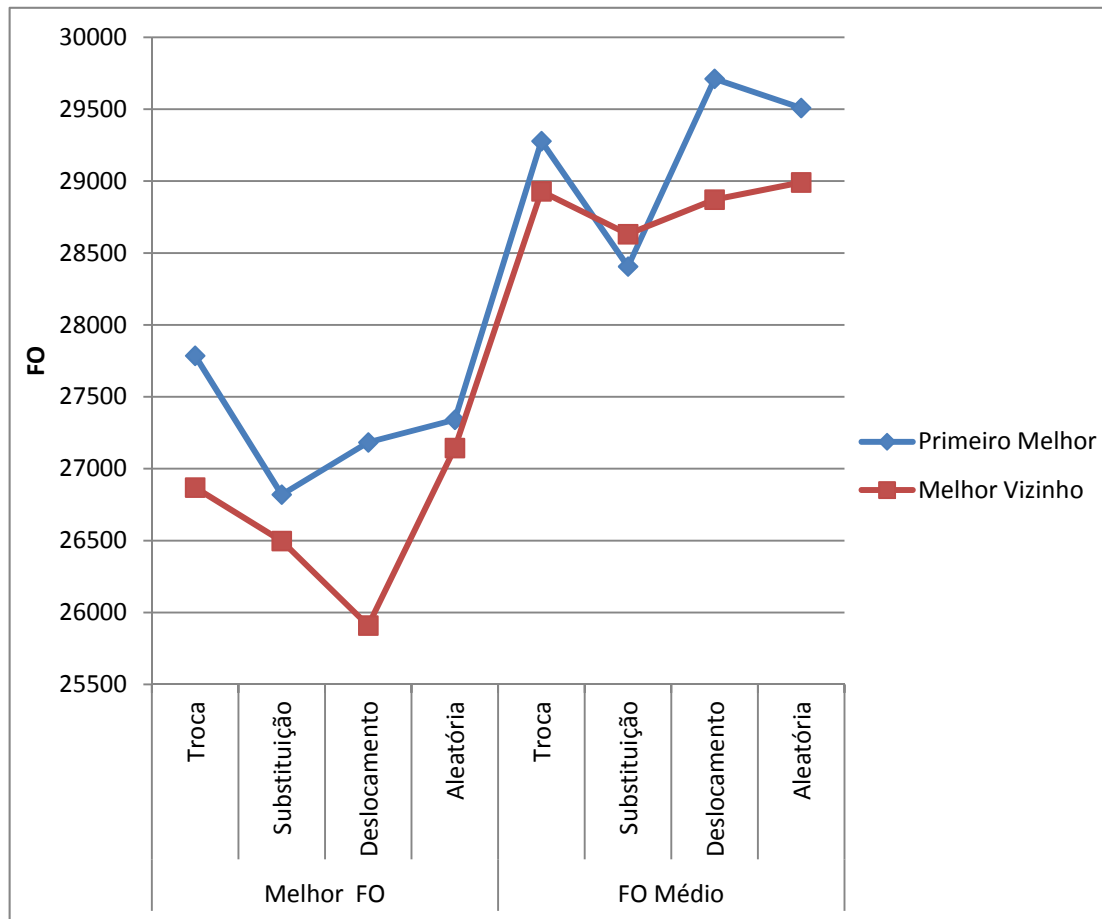


Figura 26 - Comparação entre as estratégias Primeiro Melhor e Melhor Vizinho.

A Figura 27 ilustra os dados apresentados na Tabela 3 comparando os movimentos segundo os critérios de Melhor FO e FO Médio. Nota-se que Substituição sobrepõe-se em quase todos os resultados. E quando isso não ocorre, o valor encontrado corresponde a uma única execução e portanto, deve-se considerar que o fator aleatório também pode influenciar negativamente a qualidade das soluções.

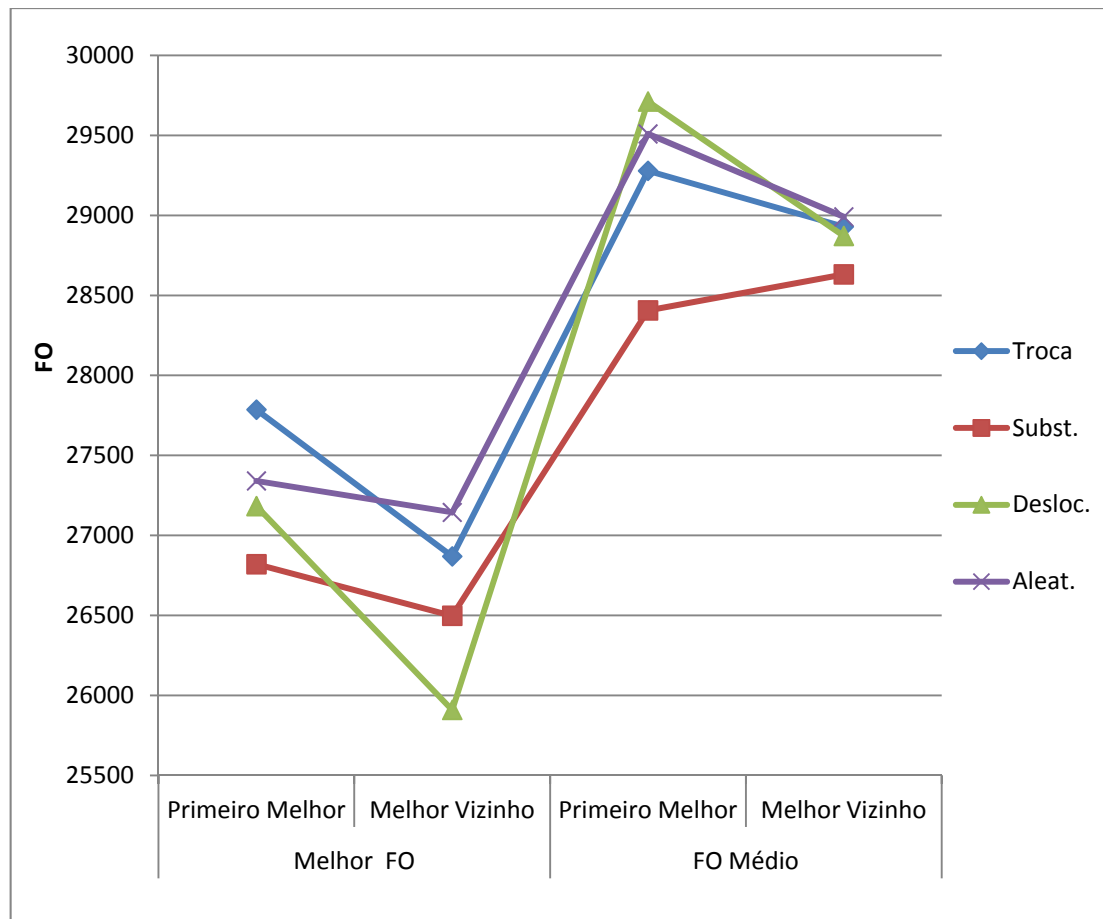


Figura 27 - Comparação entre os movimentos Troca, Substituição, Deslocamento e Aleatório.

Após a análise da Figura 26 e Figura 27, foi adotado a estratégia Melhor Vizinho combinada com movimento Substituição durante os testes computacionais apresentados nas próximas seções, inclusive na Busca Tabu.

5.3.2 *Quantidade e Capacidade dos Veículos*

Com o objetivo de identificar a influência da quantidade e capacidade dos veículos sobre os resultados encontrados, os dados são apresentados resumidamente agrupados sob os critérios de quantidade e capacidade dos veículos na Tabela 4. Percebe-se que, como esperado, o acréscimo na quantidade ou capacidade do veículo relaciona-se de maneira inversa aos valores de FO Médio. Essa informação pode ser verificada através da Figura 28 e Figura 29.

Observando a Figura 28 pode-se comprovar que ao aumentar a quantidade de veículos, a FO terá seu valor reduzido. Logo, há uma relação inversa entre eles.

Tabela 4 - Resumo dos resultados de FO Médio obtidos pelas estratégias e movimentos.

Estratégia	Movimento	Capacidade dos veículos (Q)	Quantidade de veículos (K)		
			K4	K5	K6
Primeiro Melhor	Troca	Q10	35179,7	31029,4	26714,6
		Q15	30709,9	27264,5	24771,5
	Substituição	Q10	33394,1	29696,3	26505,2
		Q15	28981,6	27227,4	24631,2
	Deslocamento	Q10	34379,5	31395,9	26973,4
		Q15	31661,1	28437,6	25422,7
	Aleatório	Q10	35312,3	30355,2	27547,1
		Q15	31292,8	27518,7	25027,3
Melhor Vizinho	Troca	Q10	34376,8	30127,6	26423,9
		Q15	30381,4	27320,2	24945,3
	Substituição	Q10	33982,3	30679,1	26867,0
		Q15	28648,4	26837,7	24774,2
	Deslocamento	Q10	34309,1	30006,6	27281,5
		Q15	29973,6	26812,7	24845,4
	Aleatório	Q10	34698,4	30502,7	26449,3
		Q15	31020,0	27025,1	24251,4

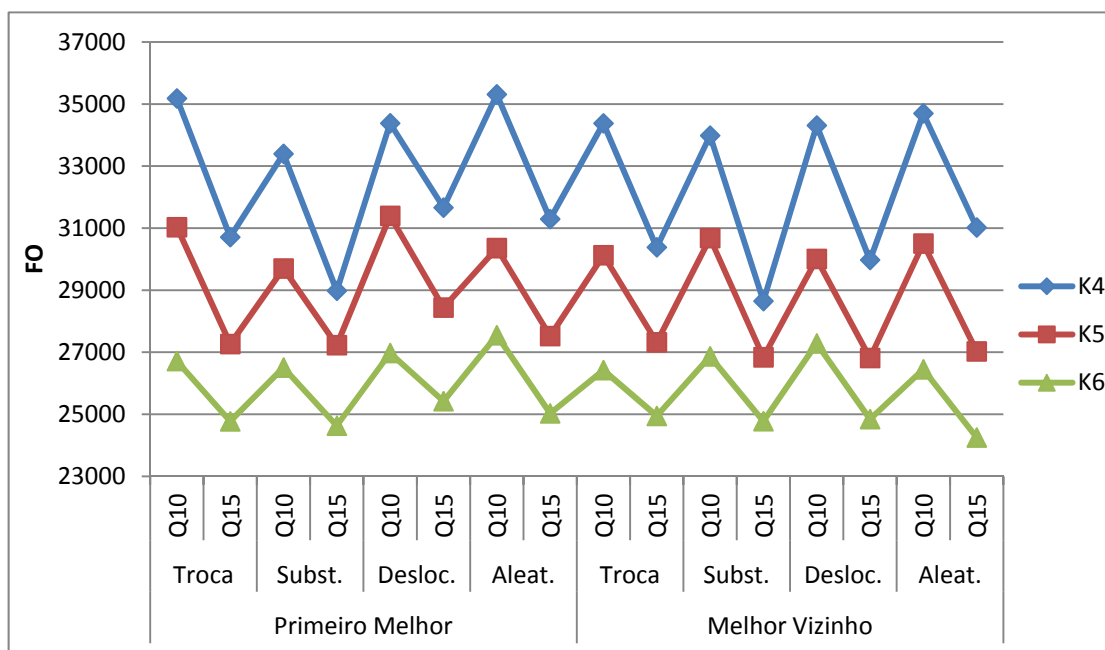


Figura 28 - Influência da quantidade de veículos sobre FO Médio.

Em relação à Figura 29, nota-se que independentemente da estratégia ou movimentos adotados, a disponibilidade de veículos com maior capacidade implica em uma redução do custo total (FO). Isso pode ocorrer por diversos fatores: maior número de solicitações atendidas, rotas mais eficientes, entre outras.

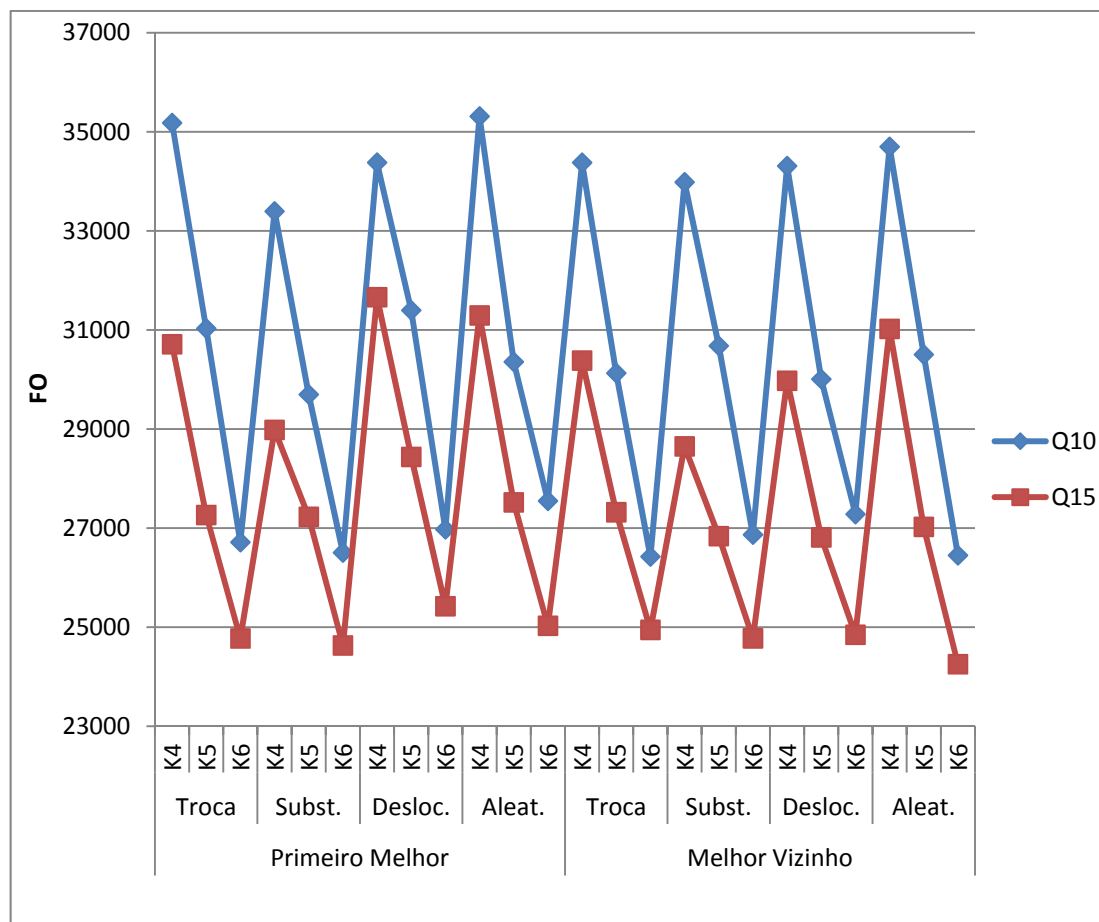


Figura 29 - Influência da capacidade do veículo sobre FO Médio.

Na Tabela 5 são apresentadas as médias dos valores de FO Médio tendo em vista apenas a quantidade e capacidade dos veículos. Observa-se que, no intervalo dos valores testados para quantidade e capacidade dos veículos, ao aumentar em um veículo a frota, há uma redução média de, pelo menos, 9% no valor de FO, enquanto o aumento da capacidade em 5 passageiros ocasiona uma redução média de pelo menos 7%.

Tabela 5 - Média dos valores de FO Médio considerando apenas Frota e Capacidade.

Quantidade de veículos (K)	Capacidade dos veículos (Q)	
	Q10	Q15
K4	34454,03	30333,60
K5	30474,10	27305,49
K6	26845,25	24833,63

A Figura 30 apresenta os dados da Tabela 5. Percebe-se que, com o aumento da quantidade de veículos, há uma tendência de queda linear de FO entre as instâncias com veículos de mesma capacidade.

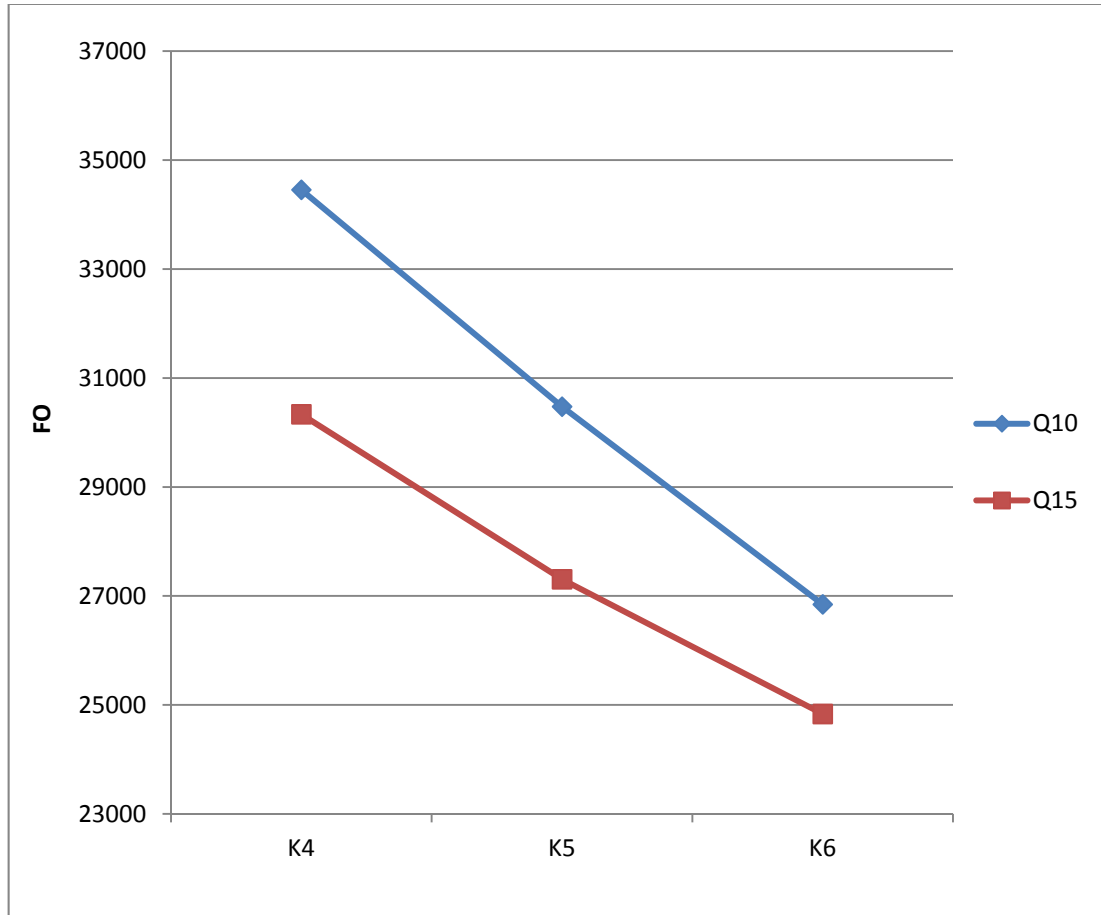


Figura 30 - Média dos valores de FO Médio considerando apenas Frota e Capacidade.

5.3.3 Métodos para o caso estático

Para cada critério (Melhor FO e FO Médio), ao comparar os resultados das heurísticas Construtiva Gulosa (Tabela 6) e GRASP-like (Tabela 7), propostas por GOMES *et al.* (2010), com o GRASP Reativo proposto (Tabela 8), percebe-se que o método GRASP Reativo teve um bom desempenho sobre as heurísticas da literatura em quase todas as instâncias. A exceção ocorreu na instância P110-K5-Q10 utilizando o critério de Melhor FO. Conforme já comentado anteriormente, devemos ser cautelosos ao julgar utilizando este critério, pois está fundamentado na observação de apenas uma execução. O GRASP-like não possuía Busca Local, mas a utilização deste procedimento no método GRASP Reativo proposto contribuiu efetivamente para o aperfeiçoamento da qualidade da solução. Analisando-se o

critério FO Médio, nota-se que este método alcançou, em média, uma redução de 10,27% e 21,72% sobre as heurísticas GRASP-like e Construtivo Guloso, respectivamente.

Tabela 6 - Resultado da heurística Construtiva Gulosa (GOMES *et al.*, 2010).

Instância	Melhor FO	FO Médio
P110-K4-Q10	39837,00	39837,00
P110-K4-Q15	39037,00	39037,00
P110-K5-Q10	34607,00	34607,00
P110-K5-Q15	37747,00	37747,00
P110-K6-Q10	32558,00	32558,00
P110-K6-Q15	35680,00	35680,00
Média	36577,67	36577,67

Tabela 7 - Resultado da heurística GRASP-like (GOMES *et al.*, 2010).

Instância	Melhor FO	FO Médio
P110-K4-Q10	35090,00	37931,60
P110-K4-Q15	32661,00	34949,90
P110-K5-Q10	28984,00	31855,40
P110-K5-Q15	28166,00	30118,50
P110-K6-Q10	27619,00	29366,30
P110-K6-Q15	25304,00	27224,30
Média	29637,33	31907,67

Tabela 8 - Resultado do método GRASP Reativo proposto.

Instância	Melhor FO	FO Médio
P110-K4-Q10	31836,00	33982,30
P110-K4-Q15	25529,00	28648,40
P110-K5-Q10	29762,00	30679,10
P110-K5-Q15	25014,00	26837,70
P110-K6-Q10	23834,00	26867,00
P110-K6-Q15	23010,00	24774,20
Média	26497,50	28631,45

A superioridade do GRASP Reativo pode ser vista mais claramente num gráfico (Figura 31) com os valores das tabelas.

A Tabela 9 foi gerada a partir do melhor resultado encontrado de cada instância para o caso estático sob os critérios de Melhor FO e FO Médio, considerando os resultados das heurísticas da literatura (Construtivo Guloso, GRASP-like) e de testes de estratégia e movimento utilizando o método GRASP Reativo. Daqui em diante, esses resultados serão denominados Melhor Resultado Encontrado.

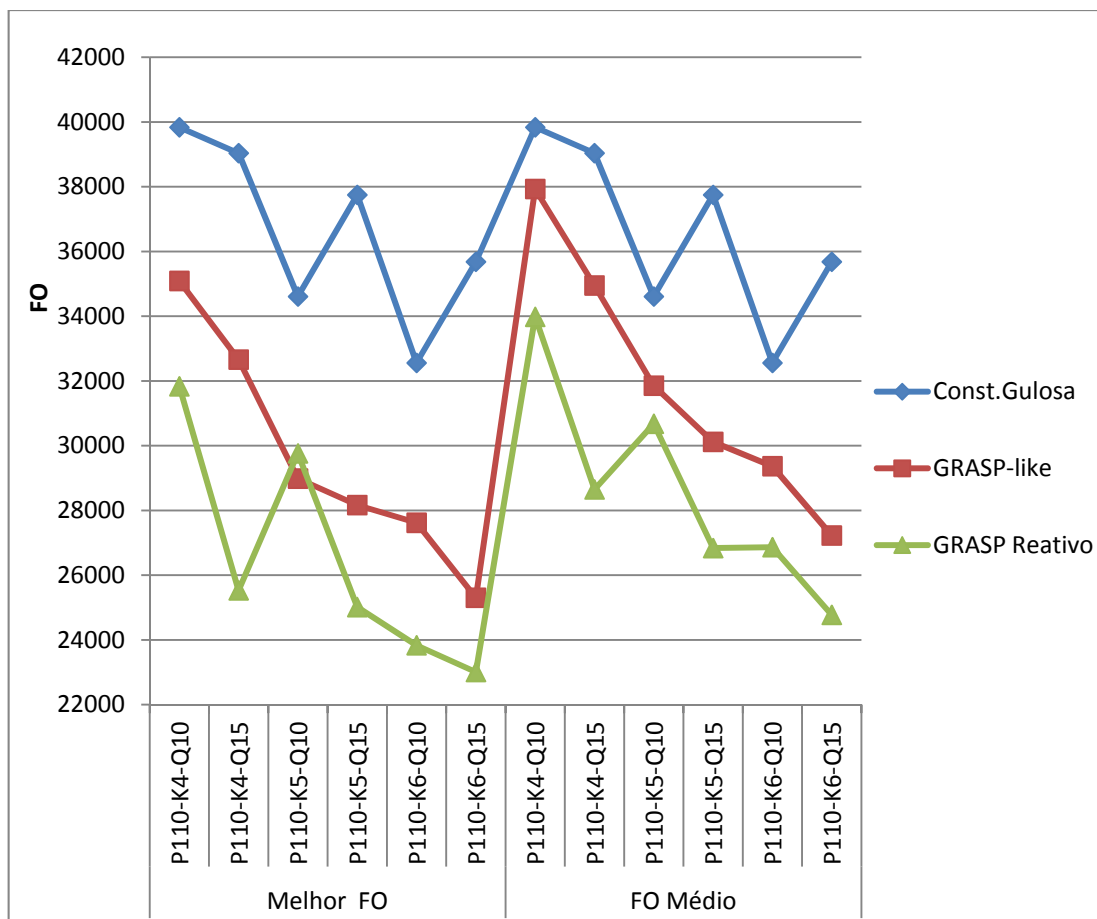


Figura 31 - Comparação entre as heurísticas Construtiva Gulosa (GOMES *et al.*, 2010), GRASP-like (GOMES *et al.*, 2010) e GRASP Reativo proposto.

Tabela 9 - Melhor Resultado Encontrado de cada instância para o caso estático sob os critérios de Melhor FO e FO Médio (não considerando GRASP Reativo + Busca Tabu).

Instância	Melhor FO	FO Médio
P110-K4-Q10	30200,00	33394,10
P110-K4-Q15	25529,00	28648,40
P110-K5-Q10	26664,00	29696,30
P110-K5-Q15	23377,00	26812,70
P110-K6-Q10	22919,00	26423,90
P110-K6-Q15	22623,00	24251,40
Média	25218,67	28204,47

Para cada critério (Melhor FO e FO Médio), ao analisar os resultados do GRASP Reativo proposto (Tabela 8), Melhor Resultado Encontrado (Tabela 9) e resultados do GRASP Reativo + Busca Tabu (Tabela 10) observa-se que o método GRASP Reativo + Busca Tabu proposto alcançou seus objetivos uma vez que conseguiu melhor desempenho que o GRASP Reativo proposto e até mesmo sobre o Melhor Resultado Encontrado de cada instância para o caso estático. Esta melhora deve-se à habilidade do método de busca em conseguir evitar mínimos locais já

visitados. Pelos dados, GRASP Reativo + Busca Tabu alcançou, em média, uma redução de 3,11% e 4,56% em relação ao Melhor Resultado Encontrado para caso estático e o método GRASP Reativo proposto, utilizando o critério de FO Médio.

Tabela 10 - Resultado do método GRASP Reativo + Busca Tabu proposto.

Instância	Melhor FO	FO Médio
P110-K4-Q10	31808,00	32368,67
P110-K4-Q15	27784,00	28616,67
P110-K5-Q10	27383,00	28593,83
P110-K5-Q15	23444,00	25402,17
P110-K6-Q10	24593,00	25649,17
P110-K6-Q15	21188,00	23332,00
Média	26033,33	27327,09

A Figura 32 foi construída com os dados das tabelas 8, 9 e 10 e ilustra o que foi comentado. Ao observar o critério FO Médio, GRASP Reativo + Busca Tabu obteve desempenho melhor ou igual ao Melhor Resultado Encontrado.

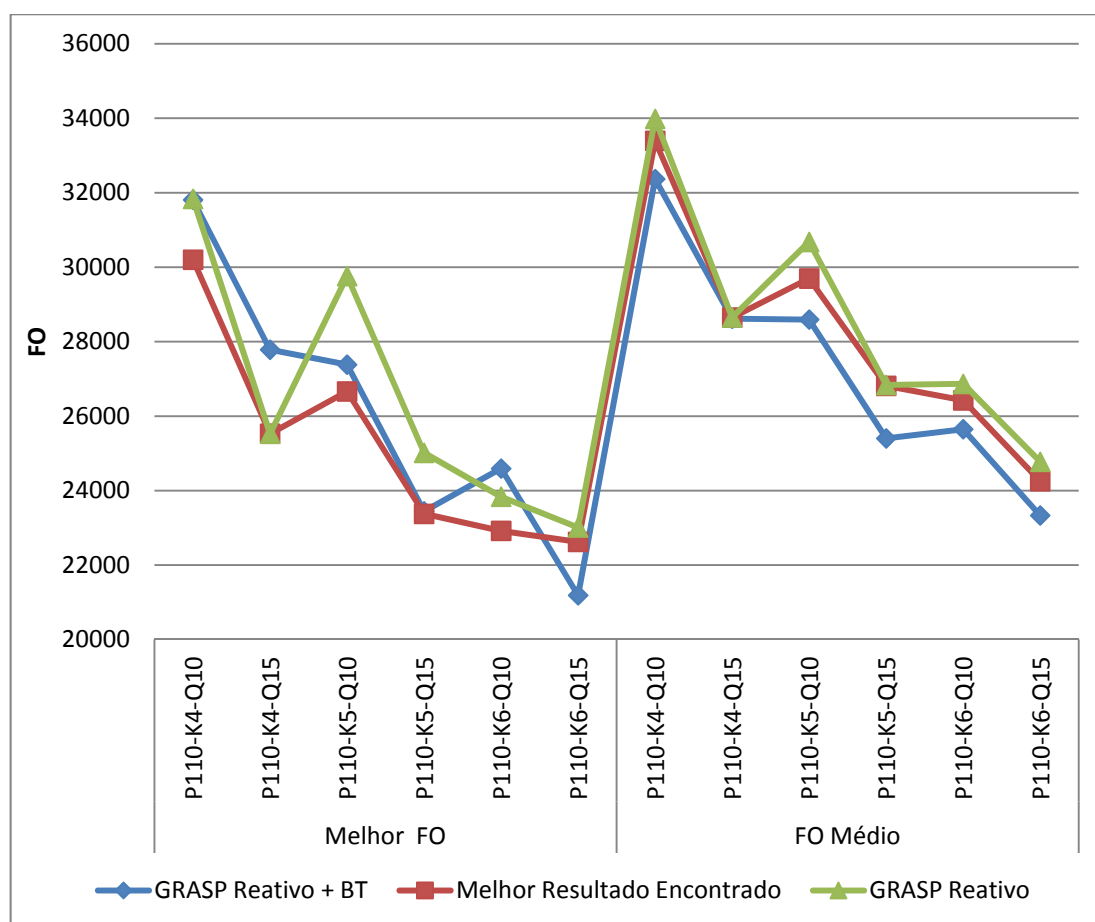


Figura 32 - Comparação entre GRASP Reativo + BT, GRASP Reativo e Melhor Resultado Encontrado de cada instância para o caso estático sob os critérios de Melhor FO e FO Médio (não considerando GRASP Reativo + BT).

5.3.4 Métodos para o caso dinâmico

Esta subseção foi organizada da seguinte forma: comparação entre solução inicial e replanejada, e comparação entre solução estática e dinâmica. A solução dinâmica consiste em se considerar inicialmente apenas parte dos dados de entrada, ou seja, apenas parte das solicitações são conhecidas; as demais são inicialmente desconhecidas; então, após um planejamento inicial ter sido feito com os dados conhecidos (solução inicial), o restante dos dados passa a ser considerado, e um replanejamento é feito (solução replanejada). Para tanto, fixa-se um certo horário, a partir do qual esse restante de dados passa a ser conhecido. Neste caso, os veículos já terão percorrido as rotas parcialmente e o restante de cada uma pode ser modificado para incluir as novas solicitações. Foram testados dois cenários: no primeiro, 20 solicitações foram conhecidas apenas depois de 1 hora e 27 minutos do início do horizonte de tempo, e no segundo 40 solicitações foram conhecidas depois de 38 minutos.

A princípio será avaliado o impacto do replanejamento em relação a solução inicial utilizada. Em seguida, será avaliada a qualidade da solução replanejada em relação à situação estática.

5.3.4.1 Comparação entre solução inicial e replanejada

Visto que a solução inicial é o planejamento realizado considerando-se apenas uma parte das solicitações, ao realizar a comparação adequada dos resultados entre esta solução e seu replanejamento é necessário considerar o custo do não atendimento das novas solicitações (*posteriori*) no FO da solução inicial antes do replanejamento, isto é, o valor da FO caso não houvesse replanejamento. Ele é obtido acrescentado-se o produto entre a quantidade de solicitações posteriores e o custo fixo pelo não atendimento de uma solicitação (Y) ao valor da solução inicial. Logo, os valores apresentados para as soluções iniciais encontradas foram ajustados segundo este critério.

Ao analisar os resultados do replanejamento proposto com 20 solicitações posteriores (Tabela 11) e 40 solicitações posteriores (Tabela 12) utilizando o resultado do método proposto GRASP Reativo + Busca Tabu como solução inicial, percebe-se o quanto o atendimento de novas solicitações impacta a solução. Além disso, quanto mais elaborado for o mecanismo de replanejamento, mais próxima a

solução replanejada será da solução obtida no caso ideal, onde todas as solicitações são conhecidas de antemão. Observe que, independentemente da quantidade de solicitações posteriores, a média do valor de FO Médio obtida pelo método de replanejamento sofreu redução quando comparada à solução inicial. Na situação com 20 solicitações posteriores, nota-se que RepGuloso reduziu em média o valor de FO em 19,91% em relação à solução inicial, enquanto RepGR, 30,36%. Quando analisados os resultados com 40 novas solicitações, observa-se que o RepGR também obteve melhor desempenho que RepGuloso: 38,96% de redução média na FO contra 29,36%.

Tabela 11 - Resultado do replanejamento proposto com 20 solicitações *a posteriori* e solução inicial do GRASP Reativo + Busca Tabu proposto utilizando FO Médio.

Instância	GRASP Reativo + Busca Tabu	RepGuloso	RepGR
P110-K4-Q10	39668,11	34487,67	30296,78
P110-K4-Q15	36348,56	29690,67	26527,22
P110-K5-Q10	36073,89	28250,11	24948,00
P110-K5-Q15	34070,00	27040,56	23116,33
P110-K6-Q10	34234,67	26209,56	22532,00
P110-K6-Q15	32676,13	24979,50	20966,63
Média	35511,89	28443,01	24731,16

Tabela 12 - Resultado do replanejamento proposto com 40 solicitações *a posteriori* e solução inicial do GRASP Reativo + Busca Tabu proposto utilizando FO Médio.

Instância	GRASP Reativo + Busca Tabu	RepGuloso	RepGR
P110-K4-Q10	46651,11	36248,67	32314,22
P110-K4-Q15	44784,11	32375,67	27167,22
P110-K5-Q10	44618,67	31685,33	26998,22
P110-K5-Q15	43905,78	31653,00	26116,56
P110-K6-Q10	44248,00	27863,33	25302,00
P110-K6-Q15	43472,00	29255,33	25495,67
Média	44613,28	31513,56	27232,31

A Figura 33 ilustra o que foi comentado anteriormente sobre a comparação entre os resultados do replanejamento proposto com 20 solicitações posteriores (Tabela 11) e 40 solicitações posteriores (Tabela 12) utilizando o resultado da heurística proposta GRASP Reativo + Busca Tabu como solução inicial. Note que o método de replanejamento RepGR obteve melhor desempenho que o RepGuloso, independentemente da quantidade de solicitações posteriores.

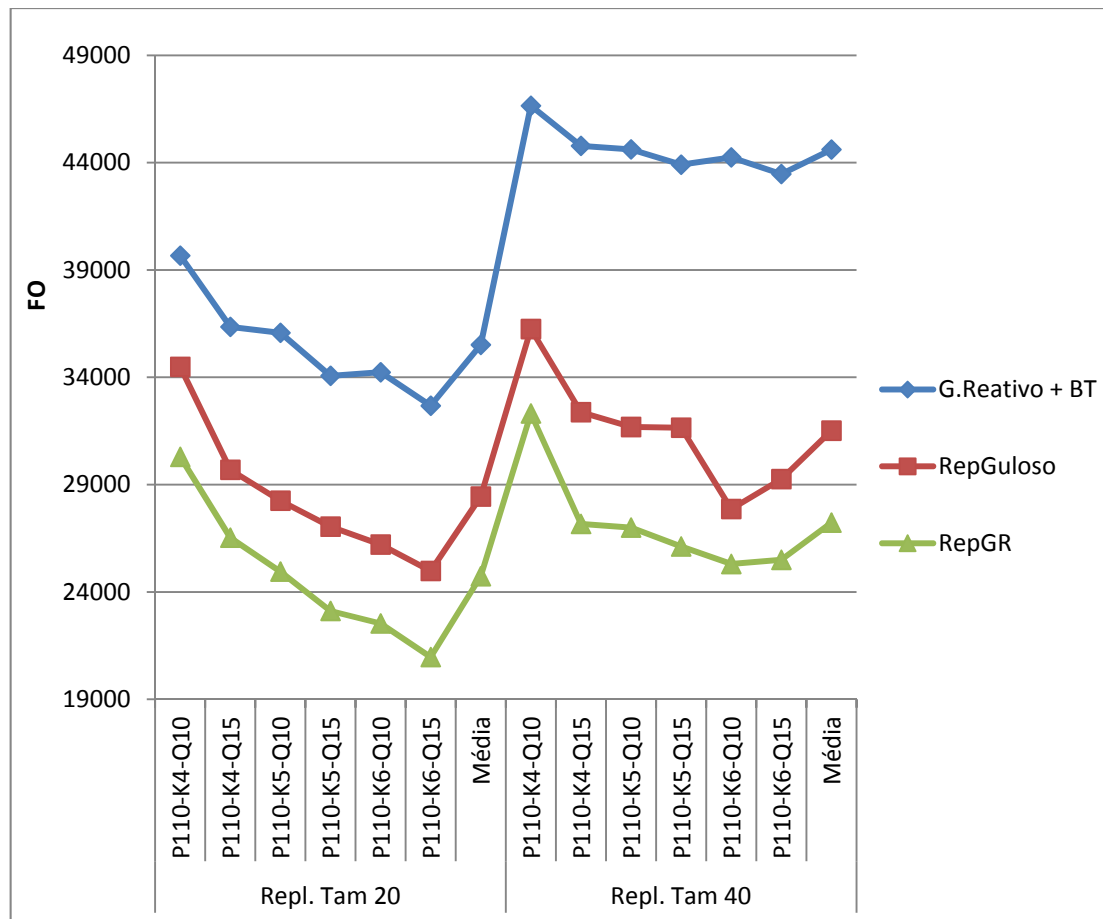


Figura 33 - Replanejamento com solução inicial da heurística GRASP Reativo + Busca Tabu.

As informações apresentadas anteriormente (Tabela 11, Tabela 12 e Figura 33) demonstram que RepGuloso e RepGR replanejam bem soluções iniciais obtidas pelo método GRASP Reativo + BT. Testes demonstraram que, de forma similar, estes métodos também replanejam de forma adequada as soluções iniciais obtidas pelas demais heurísticas (Construtiva Gulosa, GRASP-like, GRASP Reativo). Como os resultados encontrados foram semelhantes, não serão discutidos detalhadamente (esses resultados foram disponibilizados no APÊNDICE B).

A fim de identificar qual heurística utilizada como solução inicial permitiu ao método de replanejamento alcançar o melhor desempenho, os dados das médias após o replanejamento foram agrupados na Tabela 13 segundo os critérios de quantidade de solicitações posteriores e heurística inicial utilizada. Para cada combinação de método de replanejamento com quantidade de solicitações posteriores encontram-se em **negrito** os melhores resultados encontrados. Observa-se que o método de replanejamento RepGuloso, independente da quantidade de solicitações a serem atendidas posteriormente, obteve melhor desempenho médio usando como solução

inicial a heurística GRASP Reativo + BT. Já no replanejamento RepGR, houve um “empate” no desempenho médio entre GRASP Reativo e GRASP Reativo + BT como solução inicial. A heurística GRASP-*like*, mesmo obtendo o melhor resultado com 40 solicitações posteriores, não favoreceu da mesma forma o replanejamento com 20 solicitações, deixando seu desempenho médio abaixo de GRASP Reativo e GRASP Reativo + BT. Uma vez que GRASP Reativo + Busca Tabu obteve melhor desempenho médio na maioria das vezes, recomenda-se a utilização desta como solução inicial visando aumentar a eficiência da solução replanejada.

Tabela 13 - Média do resultado obtido pelos métodos de replanejamento considerando quantidade de solicitações posteriores e tipo de heurística utilizada como solução inicial.

Método Replanejamento	Qtd. Solicitações Posteriores	Heurística utilizada como solução inicial			
		Const. Gulosa	GRASP- <i>like</i>	GRASP Reativo	G. Reativo + BT
RepGuloso	20	39520,83	32102,77	29043,02	28443,01
	40	37472,67	31888,12	32104,39	31513,56
RepGR	20	36285,72	28142,63	25272,61	24731,16
	40	32214,96	26656,25	27702,22	27232,31

A Figura 34 ilustra as informações apresentadas da Tabela 13. Note que a heurística GRASP Reativo + Busca Tabu é a que mais favoreceu o método de replanejamento ao ser utilizada como solução inicial. Uma exceção ocorre em relação ao GRASP-*like* com RepGR e 40 solicitações posteriores; porém, em geral, seu desempenho pior que o GRASP Reativo.

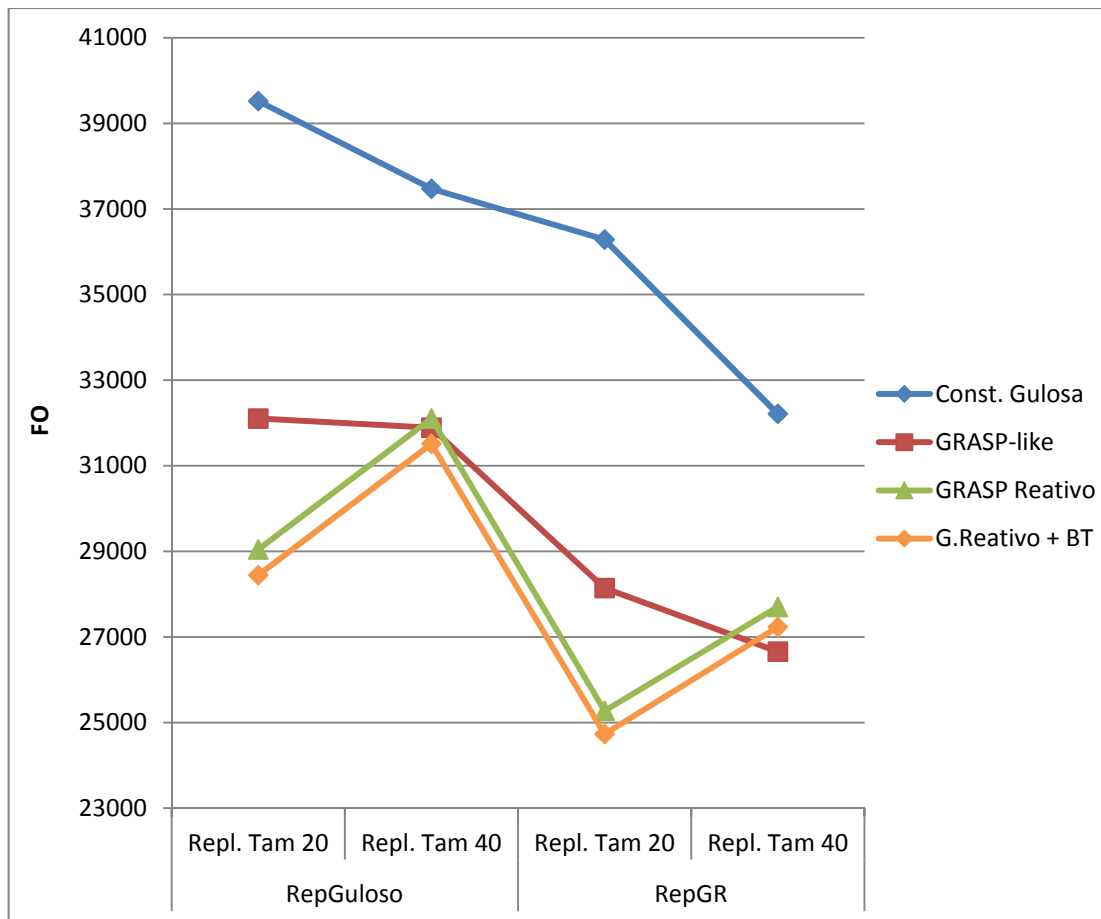


Figura 34 - Comparação entre as heurísticas utilizadas como solução inicial pelo método de replanejamento.

5.3.4.2 Comparação entre solução estática e dinâmica

A fim de confrontar os resultados do caso estático com o dinâmico, foram selecionados os dados referentes à utilização do método GRASP Reativo + BT proposto nos seguintes momentos: todas as solicitações são conhecidas de antemão (Tabela 10); replanejamento com 20 solicitações posteriores (Tabela 11) e 40 solicitações posteriores (Tabela 12). Observa-se que o replanejamento RepGR encontrou, na maioria das vezes, solução melhor do que a obtida pela aplicação da GRASP Reativo + BT no caso estático. RepGR melhorou, em média, 9,5% no tamanho 20 e 0,35% no de tamanho 40. Enquanto RepGuloso obteve melhor desempenho apenas para a instância P110-K5-Q10 com 20 solicitações posteriores. A Figura 35 ilustra esta situação.

As análises com outros métodos de solução estática e dinâmica comportam-se semelhantemente à realizada anteriormente.

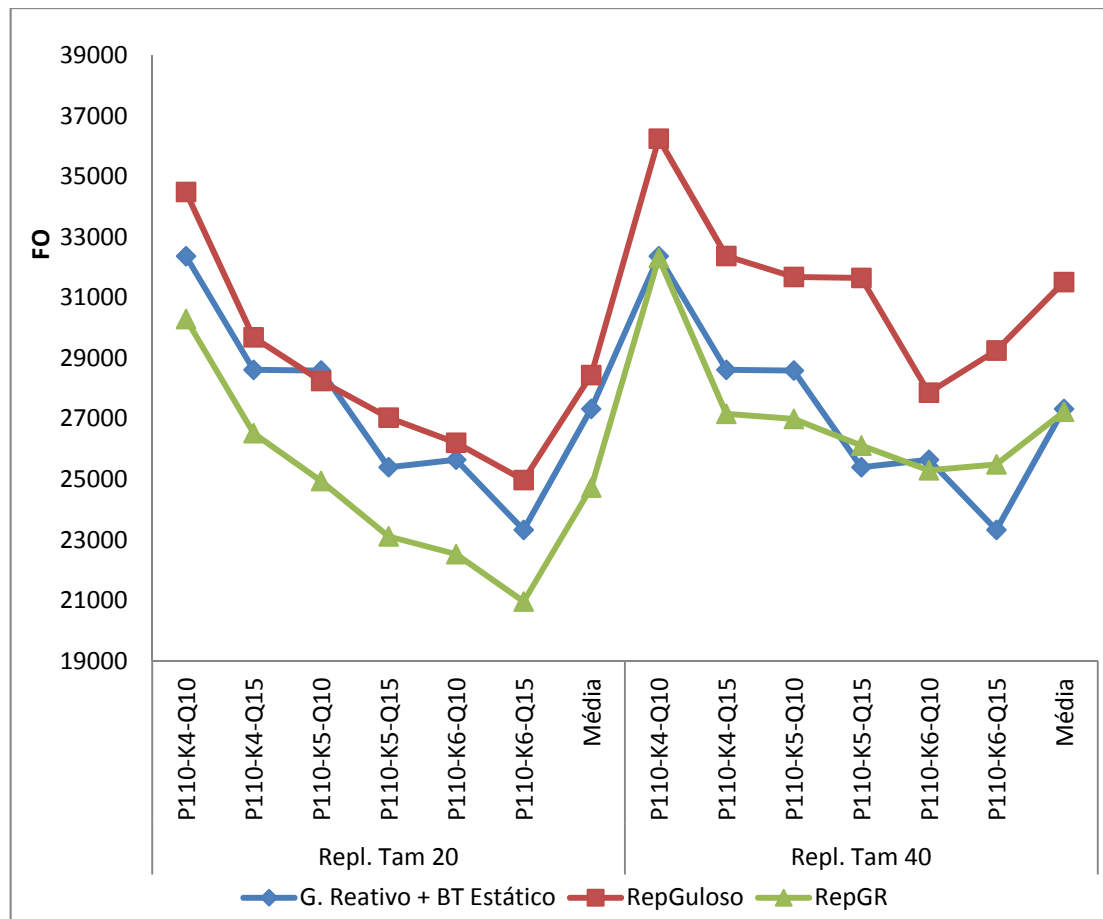


Figura 35 - Comparação entre GRASP Reativo + Busca Tabu Estático x RepGuloso e RepGR.

Com o objetivo de mensurar a eficiência do método de replanejamento do caso dinâmico em relação às heurísticas do caso estático, os dados referentes às médias dos valores obtidos foram agrupados na Tabela 14. É importante ressaltar que no caso estático todas as solicitações são conhecidas de antemão e a heurística é utilizada de forma independente. Já no caso dinâmico, há uma quantidade de solicitações que serão reveladas posteriormente e a heurística é utilizada como solução inicial para o método de replanejamento. Os valores apresentados para o caso dinâmico são as médias obtidas pelo método de replanejamento ao utilizar cada heurística. O método de replanejamento RepGuloso obteve desempenho satisfatório uma vez que alcançou valores próximos ao caso estático, onde todas as solicitações são conhecidas antecipadamente. Já o replanejamento RepGR teve um comportamento que superou as expectativas, uma vez que conseguiu soluções melhores que o caso estático. Acredito que a exploração de novas e melhores soluções foi causada pela utilização de soluções parcialmente construídas e mecanismo reativo durante o replanejamento.

Tabela 14 - Médias do resultado no caso estático e no caso dinâmico (método de replanejamento).

Heurística Utilizada	Qtd. Solicitações Posteriores	Caso Estático	Caso Dinâmico	
			RepGuloso	RepGR
Construtiva Gulosa	20	36577,67	39520,83	36285,72
	40	36577,67	37472,67	32214,96
GRASP-like	20	31907,67	32102,77	28142,63
	40	31907,67	31888,12	26656,25
GRASP Reativo	20	28631,45	29043,02	25272,61
	40	28631,45	32104,39	27702,22
GRASP Reativo + Busca Tabu	20	27327,09	28443,01	24731,16
	40	27327,09	31513,56	27232,31

A Figura 36 evidencia a situação comentada sobre a Tabela 14. Perceba que RepGuloso consegue soluções próximas ou iguais às conhecidas pelo caso Estático. Já RepGR alcançou soluções melhores ou iguais às conhecidas pelo caso Estático, independentemente da heurística utilizada como solução inicial.

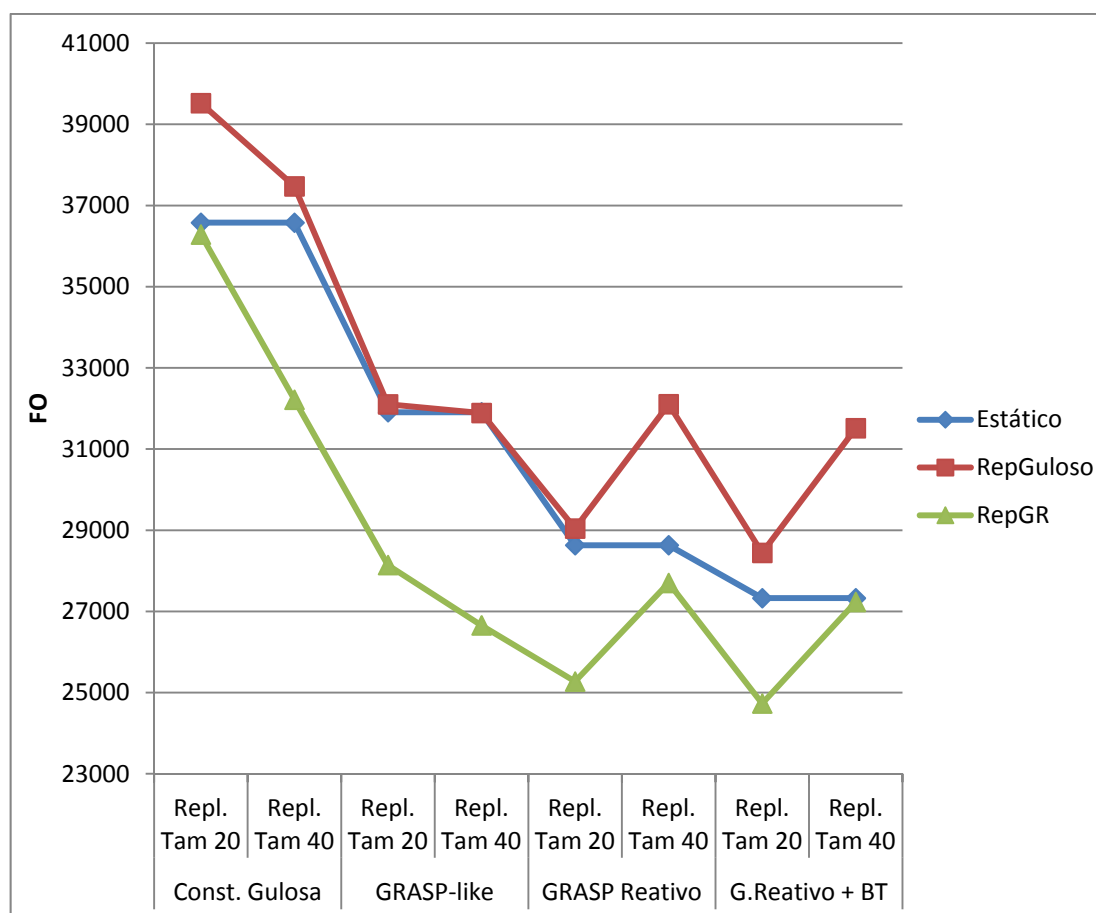


Figura 36 - Comparação entre médias do resultado obtido no caso estático e no caso dinâmico (método de replanejamento).

Após a análise da Figura 36, verificou-se que o replanejamento alcançou resultados próximos e até melhores que o método estático, o qual é aplicado conhecendo-se todas as solicitações antecipadamente. Acredito que esta superação (a qual não era esperada) se deve ao fato de gerar uma solução inicial e replanejá-la à medida em que os dados são revelados, possibilitando a exploração de soluções parciais de maneira mais eficiente (tempo, custo) e alcançar soluções que são mais atrativas e tendem a escapar de mínimos locais.

6 CONCLUSÕES

Após um levantamento bibliográfico sobre o Problema de Roteamento de Veículos Dinâmico para Transporte Reativo a Demanda (PRVDTRD), tornaram-se conhecidas as principais características do problema (consenso/compromisso entre Flexibilidade, Qualidade de Serviço e Eficiência durante o roteamento de maneira que são satisfeitas as perspectivas do passageiro e veículo) e as ferramentas disponíveis para ele (heurísticas propostas por GOMES *et al.* (2010)), bem como a necessidade de um modelo matemático completo, um conjunto de casos do problema para aplicação dos testes e ampliar este conjunto de ferramentas.

Neste trabalho foram propostos métodos de solução baseados na metaheurística GRASP Reativo e procedimento de Busca Tabu, tanto para o caso estático quando dinâmico do PRVDTRD, além de mecanismos para garantir a exploração adequada da vizinhança de uma solução e o conjunto de casos para testes.

As heurísticas propostas em (GOMES *et al.*, 2010) foram implementadas, assim como as propostas mencionadas anteriormente. Após os testes de estratégias e movimentos realizados em situação estática utilizando o GRASP Reativo, observou-se que a melhor combinação foi estratégia Melhor Vizinho com movimento Substituição. Diante desse fato, esta combinação foi a adotada durante os demais testes.

Os resultados mostraram que ao comparar soluções geradas por métodos com mecanismo reativo com as daqueles que não o possuem, o mecanismo reativo do GRASP tornou possível a utilização do parâmetro de aleatoriedade mais adequado à cada instância do problema através do processo de autocalibragem deste parâmetro, o que levou a obtenção da melhoria na qualidade das soluções obtidas.

Além disso, os resultados mostram que os métodos com mecanismo de melhoria (Busca Local, Busca Tabu) alcançaram soluções, em média, melhores quando comparados aos que não possuem este mecanismo (heurísticas propostas por GOMES *et al.* (2010)). Logo, pode-se afirmar que a utilização deste mecanismo

tornou possível explorar melhor o espaço solução, evitando assim uma convergência prematura do método.

Em relação a utilização do método de replanejamento para o caso dinâmico do problema, os resultados mostraram que é possível obter soluções com qualidade, em média, próxima ou até razoavelmente melhor que as obtidas no caso estático. Isso comprova que os métodos desenvolvidos podem ser empregados em um ambiente dinâmico, onde nem todos os dados são conhecidos *a priori*, pois consegue inserir novas solicitações de transporte às rotas previamente planejadas com baixo impacto no custo final.

Portanto, pode-se afirmar que os métodos propostos atingiram o objetivo esperado, uma vez que foi possível melhorar a qualidade da solução em relação ao proposto na literatura e garantir soluções mais próximas às necessidades dos passageiros, em situações reais, sem redução do nível de qualidade oferecido.

6.1 Trabalhos Futuros

Devido aos resultados alcançados pelo processo de melhoria neste trabalho, verifica-se a importância em se investigar outros tipos de movimentos ou critérios que favoreçam ainda mais o processo de exploração.

Além disso, há necessidade de realizar testes com outras combinações para os parâmetros utilizados e avaliar a qualidade destes estatisticamente.

Outro ponto importante é a definição e realização de testes associados a outros objetivos que podem ser minimizados além da distância total percorrida pela frota e quantidade de solicitações não atendidas, como por exemplo o tempo à bordo do veículo ou atraso em relação à janela de tempo de desembarque.

O planejamento e implementação de outras metaheurísticas, combinadas com outros métodos disponíveis na literatura pode favorecer a descoberta de soluções ainda melhores para as instâncias testadas e enriquecer ainda mais o conjunto de ferramentas disponíveis para o problema. Outra possibilidade a ser estudada é a utilização de paralelismo ao resolver o problema, visto ser uma ponderosa ferramenta computacional em que há ganhos consideráveis em relação a otimização de tempo.

Sugere-se adaptar o modelo matemático proposto por (HERBAWI; WEBER, 2012) para resolução exata do PRVDTRD e testar sua viabilidade para as instâncias utilizadas nestes trabalho, mesmo sabendo que o modelo poderá ser usado apenas no caso estático.

O teste com instâncias de outros tamanhos (quantidade de solicitações, veículos, capacidade) é imprescindível para avaliação do comportamento e o desempenho das ferramentas desenvolvidas. Algo interessante é a comparação com softwares específicos para roteamento, como por exemplo, módulo ROUTER do software LOGWARE (BALLOU, 2001, 2006), TMW Appian DirectRoute (TMW SYSTEMS INC., 2012), Roadnet Transportation Suite (ROADNET TECHNOLOGIES INC., 2012), ArcLogistics (ROUTE SOLUTIONS INC., 2012a), StreetSync Basic (ROUTE SOLUTIONS INC., 2012b), StreetSync Desktop (ROUTE SOLUTIONS INC., 2012c), JOpt.AAS (DNA EVOLUTIONS GMBH, 2012a) e JOpt.SDK (DNA EVOLUTIONS GMBH, 2012b). Mais informações sobre outros softwares relacionados, consulte (HALL, 2012).

Uma vez que o tempo despendido durante os testes computacionais foi relativamente alto, deve-se minimizá-lo através da otimização do código ou redução do número de iterações ou até mesmo abordagem paralela.

Por fim, desenvolver um ambiente em que este otimizador possa ser utilizado para gestão de transportes públicos.

REFERÊNCIAS BIBLIOGRÁFICAS

AMBROSINO, G.; NELSON, J. D.; ROMANAZZO, M. *et al.* **Demand Responsive Transport Services: Towards the Flexible Mobility Agency**. Rome, Italy: Italian National Agency for New Technologies, Energy and the Environment, 2003. p. 1-328.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos: Planejamento, Organização e Logística Empresarial**. 4. ed. Porto Alegre, RS, Brasil: Bookman, 2001.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos/Logística Empresarial**. 5. ed. Porto Alegre, RS, Brasil: Bookman, 2006.

BENT, R.; HENTENRYCK, P. VAN. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. **Transportation Science**, v. 38, n. 4, p. 515-530, nov 2004.

BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. Routing and scheduling of vehicles and crews: The state of the art. **Computers & Operations Research**, v. 10, n. 2, p. 63-211, 1983.

BRANKE, J.; DEB, K.; MIETTINEN, K.; SŁOWIŃSKI, R. (EDS.). **Multiobjective optimization: interactive and evolutionary approaches**. Berlin, Germany: Springer-Verlag, 2008.

BRAYSY, O. A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. **INFORMS Journal on Computing**, v. 15, n. 4, p. 347-368, jan 2003.

BRÄYSY, O.; GENDREAU, M. Tabu Search heuristics for the Vehicle Routing Problem with Time Windows. **TOP**, v. 10, n. 2, p. 211-237, dez 2002.

CALDAS, M. A. F.; SANTOS, D. DE A. **Uma Heurística para Roteamento de Frota de Veículos para a Distribuição de Bebidas em uma Região Urbana**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

CHRISTOFIDES, N.; BEASLEY, J. E. The period routing problem. **Networks**, v. 14, n. 2, p. 237-256, 1984.

CORDEAU, J.-F.; DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M.; SOUMIS, F. VRP with Time Windows. In: TOTH, P.; VIGO, D. (Eds.). **The vehicle**

routing problem. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. p. 157-193.

CORDEAU, J.-F.; GENDREAU, M.; HERTZ, A.; LAPORTE, G.; SORMANY, J.-S. New Heuristics for the Vehicle Routing Problem. In: LANGEVIN, A.; RIOPEL, D. (Eds.). **Logistics Systems: Design and Optimization.** New York: Springer-Verlag, 2005. p. 279-297.

CORDEAU, J.-F.; LAPORTE, G. The dial-a-ride problem: models and algorithms. **Annals of Operations Research**, v. 153, n. 1, p. 29-46, maio 2007.

CORDEAU, J.-F.; LAPORTE, G.; POTVIN, J. Y.; SAVELSBERGH, M. W. P. Transportation on demand. In: BARNHART, C.; LAPORTE, G. (Eds.). **Handbooks in Operations Research and Management Science: Transportation.** 14. ed. Amsterdam, Netherlands: Elsevier, 2007. v. 14p. 429-466.

CORDEAU, J.-F.; LAPORTE, G.; SAVELSBERGH, M. W. P.; VIGO, D. Vehicle Routing. In: BARNHART, C.; LAPORTE, G. (Eds.). **Handbooks in Operations Research and Management Science: Transportation.** 14. ed. Amsterdam, Netherlands: Elsevier, 2007. v. 23p. 367-428.

DNA EVOLUTIONS GMBH. **JOpt.AAS.** Disponível em: <<http://www.joptaas.net/>>.

DNA EVOLUTIONS GMBH. **JOpt.SDK.** Disponível em: <<http://www.dna-evolutions.com/joptsdk.html>>.

FEO, T. A.; RESENDE, M. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, n. 2, p. 109-133, 1995.

FISHER, M. L.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, v. 11, n. 2, p. 109-124, 1981.

FLEISCHMANN, B.; GNUTZMANN, S.; SANDVOS, E. Dynamic vehicle routing based on on-line traffic information. **Transportation science**, v. 38, n. 4, p. 420-433, 2004.

GHIANI, G.; GUERRIERO, F.; LAPORTE, G.; MUSMANNO, R. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. **European Journal of Operational Research**, v. 151, n. 1, p. 1-11, 2003.

GLOVER, F. Tabu Search--Part I. **INFORMS Journal on Computing**, v. 1, n. 3, p. 190-206, 1 jan 1989.

GOEL, A.; GRUHN, V. A general vehicle routing problem. **European Journal Of Operational Research**, v. 191, n. 3, p. 650-660, 2008.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos.** 2. ed. Rio de Janeiro: Elsevier, 2005. p. 518.

GOMES, R. J.; SOUSA, J. P. DE; DIAS, T. G. **A new heuristic approach for Demand Responsive Transportation Systems**. XLII Simpósio Brasileiro de Pesquisa Operacional (XLII SBPO). *Anais...* Bento Gonçalves, RS, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2010.

HALL, R. W. 2012 Vehicle Routing Software Survey. **OR/MS Today**, v. 39, n. 1, 2012.

HERBAWI, W.; WEBER, M. **The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm**. IEEE World Congress on Computational Intelligence. *Anais...* Brisbane, Australia: IEEE CEC. , 2012.

JOKINEN, J.; SIHVOLA, T.; HYYTIA, E.; SULONEN, R. **Why urban mass demand responsive transport?** IEEE Forum on Integrated and Sustainable Transportation Systems. *Anais...* Vienna, Austria: IEEE. , jun 2011.

JOSSELIN, D.; MARILLEAU, N.; LANG, C. **Modelling Dynamic Demand Responsive Transport using an Agent Based Spatial Representation**. Disponível em: <http://s4.csregistry.org/tiki-download_file.php?fileId=24>.

KAMPKE, E. H. **Metaheurísticas para o problema de programação de tarefas em máquinas paralelas com tempos de preparação dependentes da sequência e de recursos**. Dissertação (Mestrado em Ciência da Computação), Departamento de Informática, Universidade Federal de Viçosa, 2010.

KILBY, P.; PROSSER, P.; SHAW, P. **Dynamic VRPs: A Study of Scenarios. Report APES-06-1998**. Glasgow, Scotland. Disponível em: <<http://www.dcs.st-and.ac.uk/~apes/reports/apes-06-1998.ps.gz>>. , 1998.

KONTORAVDIS, G.; BARD, J. F. A GRASP for the Vehicle Routing Problem with Time Windows. **INFORMS Journal on Computing**, v. 7, n. 1, p. 10-23, jan 1995.

LENSTRA, J.; KAN, A. H. G. R. Complexity of vehicle routing and scheduling problems. **Networks**, v. 11, n. 2, p. 221–227, 1981.

LI, Z.; GUO, S.; WANG, F.; LIM, A. **Improved GRASP with Tabu Search for Vehicle Routing with Both Time Window and Limited Number of Vehicles**. (B. Orchard, C. Yang, & A. Moonis, Eds.) IEA/AIE'2004 Proceedings of the 17th international conference on Innovations in applied artificial intelligence. *Anais...* Ottawa, Canada: Springer Springer Verlag Inc. Disponível em: <http://dx.doi.org/10.1007/978-3-540-24677-0_57>. , 2004.

MACHADO, P.; TAVARES, J.; PEREIRA, F. B.; COSTA, E. **Vehicle routing problem: Doing it the evolutionary way**. (W. B. Langdon, E. Cantú-Paz, K. E. Mathias, *et al.*, Eds.) Proceedings of the Genetic and Evolutionary Computation Conference. *Anais...* New York, USA: Morgan Kaufmann. , 2002.

MADSEN, O. B. G.; LARSEN, A.; SOLOMON, M. M. **Dynamic Vehicle Routing Systems – Survey and Classification**. Sixth Triennial Symposium on

Transportation Analysis (VI TRISTAN). **Anais...** Phuket Island, Thailand: École polytechnique fédérale de Lausanne EPFL. , 2007.

MARINAKIS, Y.; MARINAKI, M. A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem. **Expert Systems With Applications**, v. 37, n. 2, p. 1446-1455, 2010.

MARINAKIS, Y.; MARINAKI, M.; DOUNIAS, G. A hybrid particle swarm optimization algorithm for the vehicle routing problem. **Engineering Applications of Artificial Intelligence**, v. 23, n. 4, p. 463-472, 2010.

MONTEMANNI, R.; GAMBARDELLA, L. M.; RIZZOLI, A. E.; DONATI, A. V. **A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System**. Second International Workshop on Freight Transportation and Logistics. **Anais...** Palermo, Italy. Disponível em: <<ftp://ftp.idsia.ch/pub/techrep/IDSIA-23-02.pdf.gz>>. , 2003.

OLIVEIRA, H. C. B. DE; CUNHA, A. S. DA; MATEUS, G. R. **Um algoritmo híbrido baseado na geração de colunas para o problema de roteamento de veículos com janela de tempo**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

PANKRATZ, G. Dynamic vehicle routing by means of a genetic algorithm. **International Journal of Physical Distribution & Logistics Management**, v. 35, n. 5, p. 362-383, maio 2005.

PRAIS, M.; RIBEIRO, C. C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. **INFORMS Journal on Computing**, v. 12, n. 3, p. 164-176, 2000.

PSARAFTIS, H. N. Dynamic vehicle routing problems. In: GOLDEN, B. L.; ASSAD, A. A. (Eds.). **Vehicle Routing: Methods and Studies**. Amsterdam: Elsevier Science, 1988. p. 223–248.

PSARAFTIS, H. N. Dynamic vehicle routing: Status and prospects. **Annals of Operations Research**, v. 61, n. 1, p. 143-164, 1995.

PUREZA, V.; LAZARIN, D. F. **Definindo rotas de atendimento para prestação de serviço em ambientes dinâmicos e estocásticos**. XLI Simpósio Brasileiro de Pesquisa Operacional (XLI SBPO). **Anais...** Porto Seguro, BA, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2009.

RIBEIRO, G. M.; LORENA, L. A. N. **Roteamento de veículos dinâmico usando algoritmos genéticos**. (A. Brasileiro, O. Lima Neto, & M. L. Maia, Eds.)XIX ANPET–Congresso de Pesquisa e Ensino em Transportes (Recife). **Anais...** Recife, PE, Brasil: Associação Nacional de Pesquisa e Ensino em Transportes – ANPET. Disponível em: <http://www.lac.inpe.br/~lorena/glaydston/Artigo_G_L_ANPET_2.pdf>. , 2005.

ROADNET TECHNOLOGIES INC. **Roadnet Transportation Suite**. Disponível em: <<http://www.roadnet.com/pub/products/Roadnet/>>.

RODRIGUES, S. B. **Metaheurística colônia de formigas aplicada a um problema de roteamento de veículos: caso da Itaipu Binacional**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

ROUTE SOLUTIONS INC. **ArcLogistics**. Disponível em: <<http://www.routesolutions.com/alr.htm>>.

ROUTE SOLUTIONS INC. **StreetSync Basic**. Disponível em: <<http://www.routesolutions.com/streetsyncbasic.htm>>.

ROUTE SOLUTIONS INC. **StreetSync Desktop**. Disponível em: <<http://www.routesolutions.com/streetsync.htm>>.

SANTOS, F. A. **Uma abordagem multiobjetivo para o problema de roteamento em redes MPLS**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

SILVA, P. DE T. G. C. **Uma variação do Push-Forward Insertion Heuristic aplicada ao Problema de Roteamento de Veículos com Janelas de Tempo**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. **Operations Research**, v. 35, n. 2, p. 254-265, 1987.

SOSA, N. G. M.; GALVÃO, R. D.; GANDELMAN, D. A. Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos. **Pesquisa Operacional**, v. 27, n. 2, p. 293-310, fev 2007.

SUBRAMANIAN, A.; OCHI, L. S.; CABRAL, L. D. A. F. **An efficient iterated local search algorithm for the vehicle routing problem with simultaneous pickup and delivery**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

TAILLARD, E.; BADEAU, P.; GENDREAU, M.; GUERTIN, F.; POTVIN, J.-Y. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. **Transportation Science**, v. 31, n. 2, p. 170-186, maio 1997.

TAILLARD, É. Parallel iterative search methods for vehicle routing problems. **Networks**, v. 23, n. 8, p. 661-673, 1993.

TMW SYSTEMS INC. **TMW Appian DirectRoute**. Disponível em: <<http://appian.tmwsystems.com/direct-route>>.

TOTH, P.; VIGO, D. **The vehicle routing problem**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. p. 363.

XAVIER, S. L. D. P.; SILVA, J. L. D. C.; SILVA, B. D. C. H. **Problema de Roteamento de Veículos aplicado à coleta de gêneros alimentícios: um estudo de caso**. XL Simpósio Brasileiro de Pesquisa Operacional (XL SBPO). **Anais...** João Pessoa, PB, Brasil: Sociedade Brasileira de Pesquisa Operacional. , 2008.

APÊNDICE A

Nesta seção são apresentados as 110 solicitações que foram utilizadas durante a realização dos testes computacionais. Conforme explicado na seção 3.1 (Definição do Problema), cada solicitação p possui as seguintes informações: identificação, vértice de embarque, vértice de desembarque, janela de tempo de embarque, janela de tempo de desembarque. Como por exemplo a solicitação `id1049_v1-v5`: `[00h06,00h09]-[01h21,01h24]`, onde sua identificação é `id1049`, vértice de embarque 1, vértice de desembarque 5, janela de tempo de embarque `[00h06,00h09]` e janela de tempo de desembarque `[01h21,01h24]`.

`id1_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1001_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1002_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1003_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1004_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1005_v5-v9`: `[00h20,00h23]-[00h50,00h53]`
`id1006_v5-v9`: `[00h21,00h24]-[00h50,00h53]`
`id1007_v5-v9`: `[00h22,00h25]-[00h50,00h53]`
`id1008_v5-v9`: `[00h23,00h26]-[00h50,00h53]`
`id1009_v5-v9`: `[00h24,00h27]-[00h50,00h53]`
`id1010_v5-v9`: `[00h25,00h28]-[00h50,00h53]`
`id1011_v2-v5`: `[00h16,00h19]-[02h40,02h43]`
`id1012_v2-v5`: `[00h16,00h19]-[02h40,02h43]`
`id1013_v2-v5`: `[00h16,00h19]-[02h40,02h43]`
`id1014_v2-v5`: `[00h16,00h19]-[02h40,02h43]`
`id1015_v2-v5`: `[00h16,00h19]-[02h40,02h43]`
`id1016_v2-v5`: `[00h17,00h20]-[02h40,02h43]`
`id1017_v2-v5`: `[00h18,00h21]-[02h40,02h43]`
`id1018_v2-v5`: `[00h19,00h22]-[02h40,02h43]`

id1019_v2-v5: [00h20,00h23]-[02h40,02h43]
id1020_v2-v5: [00h21,00h24]-[02h40,02h43]
id1021_v2-v9: [00h30,00h33]-[00h41,00h44]
id1022_v2-v9: [00h30,00h33]-[00h41,00h44]
id1023_v2-v9: [00h30,00h33]-[00h41,00h44]
id1024_v2-v9: [00h30,00h33]-[00h41,00h44]
id1025_v2-v9: [00h30,00h33]-[00h41,00h44]
id1026_v2-v9: [00h31,00h34]-[00h41,00h44]
id1027_v2-v9: [00h32,00h35]-[00h41,00h44]
id1028_v2-v9: [00h33,00h36]-[00h41,00h44]
id1029_v2-v9: [00h34,00h37]-[00h41,00h44]
id1030_v2-v9: [00h35,00h38]-[00h41,00h44]
id1031_v1-v3: [00h02,00h05]-[00h25,00h28]
id1032_v1-v3: [00h02,00h05]-[00h25,00h28]
id1033_v1-v3: [00h02,00h05]-[00h25,00h28]
id1034_v1-v3: [00h02,00h05]-[00h25,00h28]
id1035_v1-v3: [00h02,00h05]-[00h25,00h28]
id1036_v1-v3: [00h03,00h06]-[00h25,00h28]
id1037_v1-v3: [00h04,00h07]-[00h25,00h28]
id1038_v1-v3: [00h05,00h08]-[00h25,00h28]
id1039_v1-v3: [00h06,00h09]-[00h25,00h28]
id1040_v1-v3: [00h07,00h10]-[00h25,00h28]
id1041_v1-v5: [00h02,00h05]-[01h21,01h24]
id1042_v1-v5: [00h02,00h05]-[01h21,01h24]
id1043_v1-v5: [00h02,00h05]-[01h21,01h24]
id1044_v1-v5: [00h02,00h05]-[01h21,01h24]
id1045_v1-v5: [00h02,00h05]-[01h21,01h24]
id1046_v1-v5: [00h03,00h06]-[01h21,01h24]
id1047_v1-v5: [00h04,00h07]-[01h21,01h24]
id1048_v1-v5: [00h05,00h08]-[01h21,01h24]
id1049_v1-v5: [00h06,00h09]-[01h21,01h24]
id1050_v1-v5: [00h07,00h10]-[01h21,01h24]
id2_v5-v9: [00h20,00h23]-[00h50,00h53]
id3_v1-v3: [00h02,00h05]-[00h25,00h28]

id4_v1-v3: [00h07,00h10]-[00h25,00h28]
id5_v1-v8: [00h25,00h28]-[00h45,00h48]
id6_v8-v1: [00h38,00h41]-[00h58,01h01]
id7_v1-v5: [02h27,02h30]-[02h49,02h52]
id8_v2-v5: [00h16,00h19]-[02h40,02h43]
id9_v3-v10: [02h21,02h24]-[02h50,02h53]
id10_v1-v4: [00h10,00h13]-[00h32,00h35]
id11_v10-v1: [02h03,02h06]-[02h53,02h56]
id12_v5-v10: [00h08,00h11]-[00h54,00h57]
id13_v9-v11: [01h14,01h17]-[01h52,01h55]
id14_v6-v10: [02h27,02h30]-[02h39,02h42]
id15_v9-v2: [00h56,00h59]-[01h14,01h17]
id16_v10-v7: [01h00,01h03]-[02h08,02h11]
id17_v3-v9: [00h00,00h03]-[01h19,01h22]
id18_v11-v6: [01h04,01h07]-[01h50,01h53]
id19_v5-v7: [01h38,01h41]-[01h57,02h00]
id20_v1-v5: [00h02,00h05]-[01h21,01h24]
id21_v2-v5: [00h51,00h54]-[02h56,02h59]
id22_v5-v1: [01h44,01h47]-[02h00,02h03]
id23_v7-v2: [00h23,00h26]-[00h43,00h46]
id24_v6-v4: [01h44,01h47]-[02h18,02h21]
id25_v11-v7: [00h27,00h30]-[02h08,02h11]
id26_v7-v3: [02h05,02h08]-[02h46,02h49]
id27_v5-v1: [01h05,01h08]-[01h26,01h29]
id28_v6-v4: [02h06,02h09]-[02h32,02h35]
id29_v8-v1: [00h29,00h32]-[00h53,00h56]
id30_v2-v6: [02h08,02h11]-[02h32,02h35]
id31_v4-v3: [01h18,01h21]-[01h42,01h45]
id32_v9-v10: [01h03,01h06]-[01h39,01h42]
id33_v8-v3: [02h13,02h16]-[02h31,02h34]
id34_v10-v6: [01h46,01h49]-[01h59,02h02]
id35_v7-v10: [01h31,01h34]-[01h42,01h45]
id36_v6-v1: [01h01,01h04]-[02h24,02h27]
id37_v10-v6: [00h03,00h06]-[02h31,02h34]

id38_v9-v7: [01h31,01h34]-[02h36,02h39]
id39_v3-v6: [01h21,01h24]-[02h16,02h19]
id40_v6-v10: [01h08,01h11]-[02h18,02h21]
id41_v8-v1: [01h21,01h24]-[02h21,02h24]
id42_v5-v7: [01h07,01h10]-[01h32,01h35]
id43_v3-v4: [01h30,01h33]-[02h29,02h32]
id44_v8-v6: [00h52,00h55]-[01h04,01h07]
id45_v2-v9: [01h33,01h36]-[02h18,02h21]
id46_v2-v9: [00h30,00h33]-[00h41,00h44]
id47_v8-v1: [02h07,02h10]-[02h58,03h01]
id48_v5-v1: [00h33,00h36]-[02h38,02h41]
id49_v6-v8: [00h52,00h55]-[01h29,01h32]
id50_v2-v8: [01h27,01h30]-[01h52,01h55]
id51_v7-v5: [00h03,00h06]-[00h56,00h59]
id52_v9-v4: [02h04,02h07]-[02h18,02h21]
id53_v10-v8: [00h19,00h22]-[02h32,02h35]
id54_v7-v1: [00h19,00h22]-[00h44,00h47]
id55_v10-v3: [00h51,00h54]-[02h45,02h48]
id56_v5-v2: [01h06,01h09]-[02h09,02h12]
id57_v7-v3: [00h16,00h19]-[00h34,00h37]
id58_v4-v1: [01h24,01h27]-[01h49,01h52]
id59_v7-v2: [01h21,01h24]-[02h06,02h09]
id60_v6-v1: [02h13,02h16]-[02h31,02h34]

APÊNDICE B

Nesta seção são exibidos os demais resultados obtidos durante o processo de experimentação computacional não apresentados detalhadamente ao longo da análise e discussão dos resultados.

Tabela 15 - Melhor FO encontrado utilizando estratégia Primeiro Melhor.

Instância	Movimentos			
	Troca	Substituição	Deslocamento	Aleatório
P110-K4-Q10	34239,00	31891,00	30200,00	33416,00
P110-K4-Q15	29427,00	27117,00	28704,00	27895,00
P110-K5-Q10	29754,00	27425,00	29688,00	28966,00
P110-K5-Q15	25828,00	26559,00	27091,00	25815,00
P110-K6-Q10	24402,00	24522,00	22919,00	25329,00
P110-K6-Q15	23064,00	23406,00	24496,00	22623,00
Média	27785,67	26820,00	27183,00	27340,67

Tabela 16 - FO Médio utilizando estratégia Primeiro Melhor.

Instância	Movimentos			
	Troca	Substituição	Deslocamento	Aleatório
P110-K4-Q10	35179,70	33394,10	34379,50	35312,30
P110-K4-Q15	30709,90	28981,60	31661,10	31292,80
P110-K5-Q10	31029,40	29696,30	31395,90	30355,20
P110-K5-Q15	27264,50	27227,40	28437,60	27518,70
P110-K6-Q10	26714,60	26505,20	26973,40	27547,10
P110-K6-Q15	24771,50	24631,20	25422,70	25027,30
Média	29278,27	28405,97	29711,70	29508,90

Tabela 17 - Melhor FO encontrado utilizando estratégia Melhor Vizinho.

Instância	Movimentos			
	Troca	Substituição	Deslocamento	Aleatório
P110-K4-Q10	32577,00	31836,00	30234,00	32644,00
P110-K4-Q15	27081,00	25529,00	27782,00	28655,00
P110-K5-Q10	28986,00	29762,00	26664,00	28239,00
P110-K5-Q15	24989,00	25014,00	23377,00	25736,00
P110-K6-Q10	23797,00	23834,00	23762,00	24559,00
P110-K6-Q15	23784,00	23010,00	23637,00	23030,00
Média	26869,00	26497,50	25909,33	27143,83

Tabela 18 - FO Médio utilizando estratégia Melhor Vizinho.

Instância	Movimentos			
	Troca	Substituição	Deslocamento	Aleatório
P110-K4-Q10	34376,80	33982,30	34309,10	34698,40
P110-K4-Q15	30381,40	28648,40	29973,60	31020,00
P110-K5-Q10	30127,60	30679,10	30006,60	30502,70
P110-K5-Q15	27320,20	26837,70	26812,70	27025,10
P110-K6-Q10	26423,90	26867,00	27281,50	26449,30
P110-K6-Q15	24945,30	24774,20	24845,40	24251,40
Média	28929,20	28631,45	28871,48	28991,15

Tabela 19 - Resultado do replanejamento proposto com 20 solicitações a *posteriori* e solução inicial da heurística Construtiva Gulosa (GOMES *et al.*, 2010) utilizando FO Médio.

Instância	Construtiva Gulosa	RepGuloso	RepGR
P110-K4-Q10	51721,00	42277,00	40941,89
P110-K4-Q15	52445,00	44649,00	40588,33
P110-K5-Q10	50419,00	40232,00	37892,67
P110-K5-Q15	47915,00	38568,00	34182,22
P110-K6-Q10	45089,00	33351,00	31179,56
P110-K6-Q15	47395,00	38048,00	32929,67
Média	49164,00	39520,83	36285,72

Tabela 20 - Resultado do replanejamento proposto com 40 solicitações a *posteriori* e solução inicial da heurística Construtiva Gulosa (GOMES *et al.*, 2010) utilizando FO Médio.

Instância	Construtiva Gulosa	RepGuloso	RepGR
P110-K4-Q10	57837,00	46134,00	39470,89
P110-K4-Q15	53858,00	41459,00	35968,78
P110-K5-Q10	53297,00	40831,00	32767,56
P110-K5-Q15	53341,00	35395,00	32248,11
P110-K6-Q10	52780,00	32549,00	29022,33
P110-K6-Q15	47200,00	28468,00	23812,11
Média	53052,17	37472,67	32214,96

Tabela 21 - Resultado do replanejamento proposto com 20 solicitações a *posteriori* e solução inicial da heurística GRASP-like (GOMES *et al.*, 2010) utilizando FO Médio.

Instância	GRASP-like	RepGuloso	RepGR
P110-K4-Q10	44210,70	38057,90	34498,50
P110-K4-Q15	41193,20	34303,00	31061,00
P110-K5-Q10	39316,90	31828,80	28086,40
P110-K5-Q15	38748,10	31566,40	27024,30
P110-K6-Q10	36571,60	29034,30	24054,30
P110-K6-Q15	35909,50	27826,20	24131,30
Média	39325,00	32102,77	28142,63

Tabela 22 - Resultado do replanejamento proposto com 40 solicitações a *posteriori* e solução inicial da heurística GRASP-like (GOMES *et al.*, 2010) utilizando FO Médio.

Instância	GRASP-like	RepGuloso	RepGR
P110-K4-Q10	50997,60	38353,70	32705,40
P110-K4-Q15	47791,80	34939,80	29795,20
P110-K5-Q10	47229,30	31611,20	26589,60
P110-K5-Q15	45874,10	31784,80	25391,10
P110-K6-Q10	45613,30	26952,80	22100,40
P110-K6-Q15	45117,60	27686,40	23355,80
Média	47103,95	31888,12	26656,25

Tabela 23 - Resultado do replanejamento proposto com 20 solicitações a *posteriori* e solução inicial do GRASP Reativo proposto utilizando FO Médio.

Instância	GRASP Reativo	RepGuloso	RepGR
P110-K4-Q10	40514,33	34813,78	31097,67
P110-K4-Q15	36370,89	30576,33	26829,67
P110-K5-Q10	37601,78	29495,22	26190,33
P110-K5-Q15	34689,44	27411,33	23304,44
P110-K6-Q10	34943,78	26318,00	22315,89
P110-K6-Q15	33504,33	25643,44	21897,67
Média	36270,76	29043,02	25272,61

Tabela 24 - Resultado do replanejamento proposto com 40 solicitações a *posteriori* e solução inicial do GRASP Reativo proposto utilizando FO Médio.

Instância	GRASP Reativo	RepGuloso	RepGR
P110-K4-Q10	47217,67	35751,78	32726,11
P110-K4-Q15	45276,56	33344,89	29804,89
P110-K5-Q10	45721,00	33455,22	28814,89
P110-K5-Q15	44237,11	31357,00	26213,22
P110-K6-Q10	44580,22	28554,44	24119,00
P110-K6-Q15	43978,00	30163,00	24535,22
Média	45168,43	32104,39	27702,22