

Metaheurísticas Aplicadas ao Roteamento de Veículos para Transporte Reativo a Demanda

Dângelo Silva Miranda
Renan Santos Mendes
Programa de Pós-Graduação
em Modelagem Matemática
e Computacional
CEFET-MG, MG, Brasil

{dgsmiranda , renansantosmendes}@gmail.com

André Gustavo dos Santos
Departamento de Informática
Universidade Federal de Viçosa
Viçosa, MG, Brasil
andre@dpi.ufv.br

Elizabeth F. Wanner
Lab. de Algoritmos,
Metaheurísticas e Otimização
Depto de Computação
CEFET-MG, MG, Brasil
efwanner@decom.cefetmg.br

Resumo—A gestão de transportes públicos carece 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 em regiões de demanda baixa, variável e imprevisível. Dada a busca por alternativas que possibilitem o serviço sem a degradação da qualidade oferecida e não aumente os custos dos fornecedores, o problema de Roteamento de Veículos Dinâmico para Transporte Reativo a Demanda foi proposto. Neste trabalho, o problema é resolvido usando uma combinação de metaheurísticas, estratégias de busca local e de movimentos para melhorar a solução do problema. Os resultados obtidos, ao serem comparados à uma abordagem semelhante, mostram que a presente proposta apresenta melhores resultados uma vez que a solução obtida é melhor do que a solução inicialmente comparada.

Index Terms—Roteamento de Veículos, Transporte Reativo a Demanda, GRASP e Busca Tabu

I. INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV) pertence a uma classe de problemas de distribuição [1] cujo objetivo é determinar um conjunto de rotas de coleta e/ou entrega que minimize a distância total percorrida sujeito a restrições, que podem ser: tempo, tamanho da frota, demanda, entre outros.

Segundo Giani *et al.* [2], quando os dados de entrada do PRV dependem explicitamente do tempo, ele é classificado como dinâmico; caso contrário, estático. Quando todos os dados são conhecidos ao determinar a solução classifica-se em determinístico e em caso contrário, estocástico. Diante disso, citaremos cada combinação de problema resultante acompanhada de um exemplo: Estático e Determinístico (PRV capacitado); Estático e Estocástico (PRV capacitado com demandas estocásticas); Dinâmico e Determinístico (Problema do Caixeiro Viajante com tempos de viagem dependentes do horário); Dinâmico e Estocástico (PRV baseado em informação de tráfego on line).

O transporte reativo a demanda, do inglês *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, passageiros determinam os locais e horários de embarque e desembarque [3], além de outros aspectos a serem considerados. Para

Josselin *et al.* [4], um DRT 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 atender a solicitação recém-chegada de um passageiro ao sistema. 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. Por fim, Eficiência garante o atendimento das solicitações com um mínimo de veículos possíveis, respeitando as restrições de capacidade e nível de serviço aceitável para o passageiro.

Segundo Gomes *et al.* [5], no problema de Roteamento de Veículos Dinâmico para Transportes Reativos 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 a coleta e entrega simultânea 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 [6], não sendo possível determinar a melhor solução, mas o conjunto de soluções eficientes que mais se aproxima do conjunto de soluções eficientes do problema 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 [5] 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 (possivelmente sobreposição) de janelas de tempo em cada parada; 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 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 armazena as solicitações com embarque em v , $Pout_v$ as solicitações com desembarque em v e o índice de lotação m_v , a diferença entre o número de solicitações de Pin_v e o número de solicitações $Pout_v$, sendo que não há embarque ou desembarque de passageiros no depósito ($Pin_0 = Pout_0 = \emptyset$).

Cada solicitação p possui as seguintes informações: identificação, vértice de embarque, vértice de desembarque, janela de tempo de embarque, janela de tempo de desembarque. O conjunto de veículos $K = \{1, 2, \dots, k\}$, todos com mesma quantidade de assentos Q , representa a frota disponível.

II. ESTADO DA ARTE

Para resolução do PRVDTRD, encontram-se disponíveis na literatura os métodos Construtivo Guloso (*Greedy Constructive*) e GRASP-like proposto em [5]; além deste, os trabalhos [7] e [8] possuem correlação com este tema. O método Construtivo Guloso baseia-se exclusivamente no algoritmo Função de Ranqueamento de Nó (NRF) para construção de cada rota, onde a cada iteração é determinado qual o próximo nó que deverá ser inserido na rota segundo as perspectivas do veículo (distância para todos os outros nós a partir da posição atual; quantidade de passageiros a serem atendidos naqueles nós) e passageiro (quantidade de passageiros a bordo do veículo com o mesmo local de desembarque; janela de tempo dos passageiros nos nós restantes). A NRF é definida como sendo

$$\forall i \in V - \{0\}, NRF[i] = (\alpha_d \times CRL[i] + \alpha_p \times NRL[i]) + (\alpha_v \times DRL[i] + \alpha_t \times TRL[i]) \quad (1)$$

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

O algoritmo GRASP-like, utiliza a NRF para construir uma lista de candidatos (LC). Em seguida, utilizando um parâmetro de aleatoriedade α_k , constrói-se uma lista de candidatos restrita (LCR) contendo entre 1 a no máximo

$\alpha_k \times$ quantidade de elementos em LC. É importante ressaltar alguns detalhes: durante a construção de LC não é levado 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, este trabalho propõe a utilização dos seguintes métodos: GRASP Reativo, GRASP Reativo combinado com Busca Tabu, duas estratégias de busca local e quatro tipos de movimentos a serem utilizados pela busca durante exploração e melhoria da solução encontrada. Na próxima seção serão detalhados não apenas estes métodos, mas outras informações necessárias durante a construção dos mesmos.

III. DESCRIÇÃO DO PROBLEMA E METODOLOGIA

Cada solução S é composta por um conjunto de rotas R_i , isto é, $S = \{R_1, \dots, R_n\}$, $n \leq k$, formado a partir do conjunto de vértices V que respeita 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 vértice de embarque e desembarque de pelo menos uma solicitação de transporte $\min R = \langle 0, p^+, p^-, 0 \rangle$, onde p^+ representa o vértice de embarque; p^- representa o vértice de desembarque. Uma vez que é necessário respeitar as restrições do problema, um veículo k poderá retornar a um vértice já visitado para atendimento de alguma solicitação, porém a repetição do vértice não poderá ser consecutiva.

O custo de cada rota $R_i = \langle v_0, v_1, \dots, v_m, v_{m+1} \rangle$ é

$$C(R_i) = \sum_{j=0}^m (c_{v_j, v_{j+1}}) + W \quad (2)$$

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

$$C(U) = \sum_{j \in U} Y \quad (3)$$

em que Y é o custo fixo pelo não atendimento da solicitação.

Portanto, o custo total da solução S é:

$$C(S) = \sum_{i=1}^n C(R_i) + C(U) \quad (4)$$

A Função Objetivo (FO) que quantifica a qualidade de uma solução S foi a mesma utilizada no trabalho [5] e será dada por $C(S)$.

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*), proposta em [9], é um método onde a cada iteração é construída uma solução viável para o problema

desejado utilizando um parâmetro de aleatoriedade (α) fixo seguido de uma fase de melhoria desta solução, armazenando a melhor solução encontrada até o momento. Este método é composto por duas fases: fase de construção e fase de busca local. A cada iteração durante a fase de construção, uma lista de candidatos (LC) ordenada segundo algum critério, contém todos os elementos possíveis a serem adicionados na solução. Em seguida, utilizando o parâmetro de aleatoriedade (α), determina-se uma lista de candidatos restrita (LCR). Por fim, um candidato é escolhido aleatoriamente de LCR e será adicionado à solução em construção. Repete-se o processo até que uma solução completa seja obtida. Esta solução é melhorada durante a fase de busca local utilizando movimentos que permitem a exploração das soluções vizinhas àquela.

Na fase de construção, a solução viável (conjunto de rotas) é construída a partir da aplicação do algoritmo NRF proposto em [5], adicionando em cada rota inicialmente vazia um elemento (vértice) por vez. Em cada iteração do algoritmo NRF, uma lista de candidatos (LC) de elementos viáveis a ser inserida na rota corrente é construída, ordenada decrescentemente a partir do valor NRF atribuído àquele elemento. Elemento viável é todo aquele que ainda possuir solicitação de embarque e/ou desembarque, satisfazendo as restrições de capacidade do veículo e janela de tempo de embarque. Quanto maior for o valor NRF atribuído, mais interessante torna-se a utilização do elemento pois baseia-se na ponderação entre as perspectivas do veículo e passageiro. Em seguida, uma certa quantidade de primeiros melhores elementos serão selecionados para compor a lista de candidatos restritos (LCR). O tamanho ou quantidade de elementos desta lista ($|LCR|$) é determinado por

$$|LCR| = \max(1, \alpha_k \times |LC|) \quad (5)$$

em que $|LC|$ é o tamanho ou quantidade de elementos de LC, α_k é um parâmetro de aleatoriedade. O valor α_k é selecionado de um conjunto de possíveis α_k (A) com probabilidade de escolha P_k que é recalculada automaticamente conforme será explicado posteriormente. Como o PRVDTRD é um problema com restrições conflitantes, esta última abordagem contempla a natureza adaptativa de cada iteração do algoritmo, a variação do tamanho de LC que, às vezes, tem poucos elementos. Em seguida o algoritmo seleciona aleatoriamente um elemento de LCR e insere na rota a ser construída. No momento em que uma rota não pode satisfazer alguma solicitação de transporte, esta é finalizada. Havendo alguma solicitação insatisfeita deixada e outros veículos disponíveis, nova rota para outro veículo é iniciada e construída da mesma forma. Repete-se o processo até que não haja solicitações de transportes viáveis deixadas a satisfazer ou nenhum veículo disponível. A solução final é composta por um conjunto de rotas resultante e seu custo é calculado, sendo que esta será utilizada na fase de Busca Local.

O método GRASP Reativo, proposto em [10], 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. Desta forma, os 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 ($\frac{1}{v}$). 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 [11].

Para a fase de Busca Local foram desenvolvidos e utilizados os seguintes tipos de movimento: Troca, Substituição, Deslocamento e Aleatória. O movimento é aplicado sobre um vetor que é resultado da concatenação das rotas de uma dada solução, retirando-se o depósito.

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.
- Substituição: consiste na escolha de uma posição deste vetor e a substituição do conteúdo deste por outro, contido no conjunto de vértices de parada $V - \{0\}$.
- 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.
- 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 $|R|$, os movimentos Troca e Deslocamento geram um máximo de $\frac{|R| \times (|R|-1)}{2}$ e $(|R|-1)^2$ vizinhos, respectivamente. Este total é 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.

O procedimento de busca local 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).

A Busca Tabu (BT), proposta em [12], 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. O algoritmo BT foi utilizado na fase de busca local do GRASP Reativo.

A estratégia de Melhor Vizinho combinada ao movimento de Substituição foi implementada como estratégia de exploração de vizinhança na Busca Tabu. Razões para tal escolha serão mostradas posteriormente. 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 $listaTabu[no][posicao]$, onde $posicao \in \{0, 1, \dots, |R|-1\}$ e $nó \in V - \{0\}$.

IV. RESULTADOS COMPUTACIONAIS

Nesta seção 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. 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 [5]: 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 de [13]. Os resultados presentes em [5] serão, portanto, usados como comparação para os resultados obtidos neste trabalho. É importante salientar que para comparação do GRASP Reativo proposto com os métodos apresentados na literatura, os métodos do trabalho [5], a 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.

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).

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. 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 $NRFF$, foram os mesmos apresentados em [5]: $\alpha_d = 0.20$; $\alpha_p = 0.15$; $\alpha_v = 0.55$; $\alpha_t = 0.10$.

A função objetivo $C(S)$ 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).

O critério de parada utilizado no GRASP Reativo foi o número máximo de iterações ($MAX_ITER = 600$). 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 $\gamma = 20$.

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

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.

Na Tabela I são apresentados a média dos valores de FO Médio para cada combinação, sendo que estão em negrito o melhor resultado encontrado para cada estratégia. Testes considerando um fator por vez, tipo de movimento e tipo de estratégia, indicaram que a estratégia Melhor Vizinho e o movimento de Substituição representavam as melhores possibilidades dentro de cada fator. Desta forma, a combinação estratégia Melhor Vizinho com o movimento Substituição foi utilizada nos testes subsequentes. Não foi realizado um experimento fatorial para avaliar a interação entre os fatores considerados.

Tabela I
MÉDIA DOS VALORES DE FO OBTIDOS CONSIDERANDO APENAS ESTRATÉGIA E MOVIMENTO.

Estratégia	Movimento	FO Médio
Primeiro Melhor	Troca	29278,27
	Substituição	28405,97
	Deslocamento	29711,70
	Aleatório	29508,90
Melhor Vizinho	Troca	28929,20
	Substituição	28631,45
	Deslocamento	28871,48
	Aleatório	28991,15

Com o objetivo de identificar se há influência da quantidade e capacidade dos veículos, além da interação destes sobre os resultados encontrados, foi aplicado um Planejamento Fatorial Completo.

O procedimento estatístico conhecido como Análise de Variância (ANOVA) com dois fatores foi usado para identificar as diferenças nos valores obtidos para o problema quando os fatores Capacidade (Q) e Quantidade de Veículos (K) eram variados. Com os resultados obtidos, pode-se concluir que tanto os fatores K e Q quanto a interação entre eles ($K \times Q$) possuem efeitos significativos, indicando que existem diferenças significantes nos resultados obtidos.

Por fim, para detectar o tipo de interação entre os fatores, um Planejamento Fatorial 2^k foi utilizado. Uma vez que este planejamento necessita apenas de dois níveis para cada fator, selecionou-se os níveis 4 e 6 (o mais baixo e o mais alto) para o fator Quantidade de Veículos e todos os níveis para o fator Capacidade (10 e 15 lugares).

Novamente, as observações realizadas sobre a ANOVA para o experimento anterior aplicam-se semelhantemente a este: tanto os fatores quanto a interação são significativos. Tal comportamento era esperado ocorrer uma vez que o problema é o mesmo.

Os efeitos principais foram calculados e observou-se que ambos os fatores são inversamente proporcionais ao valor da FO, isto é, o aumento de um nível para qualquer fator (Capacidade ou Quantidade de Veículos) implica em uma redução do valor da FO. Além disso, percebe-se que, para qualquer capacidade, o aumento da quantidade de veículos na frota proporciona um grande efeito redutor sobre a FO. Por fim observa-se que veículos com menor capacidade alcançam maiores reduções de FO (visto pela inclinação) decorrente a um aumento de capacidade.

Diante dos experimentos realizados conclui-se que uma configuração que proporcionaria maior redução de custo é a utilização de seis veículos com capacidade de quinze lugares em cada.

Para cada critério (Melhor FO e FO Médio), ao comparar os resultados das heurísticas Construtiva Gulosa e GRASP-like, propostas em [5], com o GRASP Reativo proposto (Tabela II), 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.

Para cada critério (Melhor FO e FO Médio), ao analisar os resultados do GRASP Reativo proposto e resultados do GRASP Reativo + Busca Tabu (Tabela II) observa-se que o método GRASP Reativo + Busca Tabu proposto alcançou seus objetivos uma vez que conseguiu melhor desempenho que o GRASP Reativo proposto. Esta melhora deve-se à habilidade do método de busca em conseguir evitar mínimos

Tabela II
 RESULTADO DAS HEURÍSTICAS CONSTRUTIVA GULOSA [5], GRASP-like [5] E MÉTODOS PROPOSTOS GRASP REATIVO E GRASP REATIVO + BUSCA TABU.

Instância	Construtiva Gulosa		GRASP-like		GRASP Reativo		GRASP Reativo + Busca Tabu	
	Melhor FO	FO Médio	Melhor FO	FO Médio	Melhor FO	FO Médio	Melhor FO	FO Médio
P110-K4-Q10	39837,00	39837,00	35090,00	37931,60	31836,00	33982,30	31808,00	32368,67
P110-K4-Q15	39037,00	39037,00	32661,00	34949,90	25529,00	28648,40	27784,00	28616,67
P110-K5-Q10	34607,00	34607,00	28984,00	31855,40	29762,00	30679,10	27383,00	28593,83
P110-K5-Q15	37747,00	37747,00	28166,00	30118,50	25014,00	26837,70	23444,00	25402,17
P110-K6-Q10	32558,00	32558,00	27619,00	29366,30	23834,00	26867,00	24593,00	25649,17
P110-K6-Q15	35680,00	35680,00	25304,00	27224,30	23010,00	24774,20	21188,00	23332,00

locais já visitados. Pelos dados, GRASP Reativo + Busca Tabu alcançou, em média, uma redução de 4,56% em relação ao método GRASP Reativo proposto, utilizando o critério de FO Médio.

V. CONCLUSÃO

Neste trabalho foram propostos métodos de solução baseados na metaheurística GRASP Reativo e no procedimento de Busca Tabu para o caso estático 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 [5] 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 combinação da estratégia Melhor Vizinho com movimento de Substituição apresentou melhores resultados. 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. Tais resultados se aplicam apenas às instâncias artificiais usadas para o problema e novos testes precisam ser feitos com problemas de maior significância.

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 em [5]). Logo, pode-se afirmar que a utilização deste mecanismo tornou possível explorar melhor o espaço solução, evitando assim uma convergência prematura do método. Uma análise estatística mais detalhada é necessária para validar a significância dos resultados obtidos.

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.

AGRADECIMENTOS

Os autores gostariam de agradecer às agências de fomento CAPES, CNPq e FAPEMIG.

REFERÊNCIAS

- [1] N. G. M. Sosa, R. D. Galvão, and D. A. Gandelman, "Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos," *Pesquisa Operacional*, vol. 27, no. 2, pp. 293–310, Feb. 2007.
- [2] G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno, "Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies," *European Journal of Operational Research*, vol. 151, no. 1, pp. 1–11, 2003.
- [3] G. Ambrosino, J. D. Nelson, M. Romanazzo, D. Begg, M. Benedetti, C. Binazzi, M. Boero, N. D. Volo, P. Eloranta, D. Engels, A. Ferrari, B. Finn, M. Gorini, S. Grosso, A. Iacometti, L. Jussiant, I. Kaisto, A. Kalliomäki, M. K. Grimes, J. F. Mageean, B. Masson, F. Pettinelli, A. Perugia, P. Sassoli, J. Scholliers, L. Setti, G. Speroni, A. Ståhl, M. Talluri, G. Valenti, Y. Westerlund, and L. Weyns, *Demand Responsive Transport Services: Towards the Flexible Mobility Agency*, G. Ambrosino, J. D. Nelson, and M. Romanazzo, Eds. Rome, Italy: Italian National Agency for New Technologies, Energy and the Environment, 2003.
- [4] D. Josselin, N. Marilleau, and C. Lang, "Modelling Dynamic Demand Responsive Transport using an Agent Based Spatial Representation," 2008.
- [5] R. J. Gomes, J. P. de Sousa, and T. G. a. Dias, "A new heuristic approach for Demand Responsive Transportation Systems," in *XLII SBPO*. Bento Gonçalves, Brasil: Sociedade Brasileira de Pesquisa Operacional, 2010.
- [6] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds., *Multiobjective optimization: interactive and evolutionary approaches*. Berlin, Germany: Springer-Verlag, 2008.
- [7] R. Gomes, J. P. de Sousa, and T. G. a. Dias, "A GRASP-based Approach for Demand Responsive Transportation," *International Journal of Transportation*, vol. 2, no. 1, pp. 21–32, Apr. 2014.
- [8] K. Wong, A. Han, and C. Yuen, "On dynamic demand responsive transport services with degree of dynamism," *Transportmetrica A: Transport Science*, vol. 10, no. 1, pp. 55–73, Jan. 2014.
- [9] T. A. Feo and M. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.
- [10] M. PRAIS and C. RIBEIRO, "Reactive GRASP: An application to a matrix decomposition problem in tdma traffic assignment," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 164–176, 2000.
- [11] E. H. Kampke, "Metaheurísticas para o problema de programação de tarefas em máquinas paralelas com tempos de preparação dependentes da sequência e de recursos," Dissertação, Universidade Federal de Viçosa, 2010.
- [12] F. Glover, "Tabu Search—Part I," *INFORMS Journal on Computing*, vol. 1, no. 3, pp. 190–206, Jan. 1989.
- [13] D. S. Miranda, "Aplicação de metaheurísticas para o problema de roteamento de veículos dinâmico para transporte reativo a demanda," Dissertação, Universidade Federal de Viçosa, 2012.