

CommuteVRP: otimização de serviços privados de transporte contínuo

CommuteVRP: private commute transportation services optimization

Ednaldo Oliveira Lima
Universidade Federal de Lavras
Campus Universitário/DCC
Lavras, Brasil
eednaldo@gmail.com

André Vital Saúde
Universidade Federal de Lavras
Campus Universitário/DCC
Lavras, Brasil
saude@dcc.ufla.br

RESUMO

Mobilidade urbana é tema que está sempre em alta desde o surgimento das grandes metrópoles. Dentre os diversos fatores geradores de tráfego, podemos destacar os polos geradores de viagens (PGVs), locais que concentram os destinos ou origens de grande quantidade de pessoas, como centros comerciais, parques industriais e grandes condomínios. O objetivo deste trabalho é definir o Problema de Roteamento de Veículos para o Transporte Contínuo (CommuteVRP, do inglês *Commute Vehicle Routing Problem*), que descreve o deslocamento diário de pessoas a um PGV, e verificar se algoritmos já propostos na literatura são capazes de solucionar o problema com instâncias de escala próxima à demanda real do mercado. Definimos o CommuteVRP como uma variante dos problemas de roteamento de veículos que busca pela criação de rotas economicamente lucrativas com conforto e preços acessíveis aos passageiros. Trata-se de um problema de otimização diferente dos problemas clássicos de roteamento de veículos e transporte de pessoas encontrados na literatura, pois tem como foco a otimização do ponto de vista mercadológico, cuja solução é composta de rotas economicamente viáveis e lucrativas que não necessariamente atendam a todas as pessoas envolvidas. Foi possível solucionar o problema em tempo razoável utilizando a heurística do princípio *ruin and recreate* que nos permite a computação paralela para a geração de rotas. Foram alcançados resultados lucrativos para todas as instâncias testadas, e foram comparadas diferentes configurações de execução.

Palavras-Chave

Mobilidade Urbana, Problema de Roteamento de Veículos, Cidades Inteligentes, Transporte Contínuo.

ABSTRACT

Urban mobility has always been on the rise since the emergence of large metropolises. Among the several factors that

generate traffic, we can highlight the Traffic Generation Poles (TGP), places that concentrate the destinations or origins of a large number of people, such as malls, industrial parks and large condominiums. The purpose of this work is to define the CommuteVRP, *Commute Vehicle Routing Problem*, which describes the daily movement of people to a TGP, and to verify if algorithms already proposed in the literature are capable of solving this problem with instances of scale close to the actual market demand. We defined CommuteVRP as a variant of vehicle routing problems that seeks to create economically profitable routes with comfort and affordability to passengers. It is a different optimization problem from the classic problems of vehicle routing and people transportation found in the literature, since it focuses on the optimization from a marketing point of view, whose solution is composed of economically viable and profitable routes that do not necessarily meet to all the people requests. It was possible to solve the problem in a reasonable time using the *ruin and recreate* heuristic that allows parallel computing to generate routes. Profitable results were achieved for all the tested instances, and different execution configurations were compared.

CCS Concepts

•Applied computing → Transportation; Marketing;
•Theory of computation → Network flows; •Information systems → Location based services;

Keywords

Urban Mobility, Vehicle Routing Problem, Smart Cities, Commute Transportation.

1. INTRODUÇÃO

De acordo com Rodrigues [?], um sistema eficiente de mobilidade é essencial para o acesso ao mercado de trabalho, à educação, ao consumo e ao lazer, ou seja, é uma condição fundamental para a construção do bem-estar urbano. Segundo a Organização das Nações Unidas (ONU) [?], hoje 54% da população mundial vive em áreas urbanas, e, até 2050 este número deve aumentar para 66%.

Grande parte da demanda por mobilidade nas cidades é para fins de deslocamento de pessoas a partir das suas residências para seus locais de trabalho ou estudo. O transporte de pessoas em deslocamentos periódicos e regulares, entre a residência e um local de trabalho ou estudo, é de-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBSI 2017 June 5th – 8th, 2017, Lavras, Minas Gerais, Brazil

Copyright SBC 2017.

nominado *Transporte Contínuo* (*Commuting*, em inglês). O termo *Transporte Contínuo* é também utilizado para se referir a qualquer deslocamento regular ou repetitivo entre duas localidades, mesmo que sem finalidade de trabalho.

A demanda por transporte contínuo tem destaque em áreas como universidades, centros comerciais e indústrias. Essas áreas chamadas Polos Geradores de Viagens (PGV). Os PGVs causam grandes impactos na mobilidade urbana, atingindo regiões específicas e propagando-se ao seu entorno. O transporte coletivo público não consegue suprir de forma satisfatória esta demanda por mobilidade e os investimentos em infraestrutura necessários para sanar este problema seriam elevados e não sustentáveis. Segundo Scaringella [?], em países menos desenvolvidos existe uma supervalorização da ideia de se utilizar grandes obras para resolver o problema de mobilidade, e na verdade, a prioridade deve ser a racionalização do uso da infraestrutura já existente.

Nos últimos anos, o melhor aproveitamento do transporte privado vem se destacando no cenário mundial. *Startups* que gerenciam aplicativos de caronas, compartilhamento de carros (*car pooling*) ou que permitem que pessoas prestem serviços como motoristas, estão se tornando empresas milionárias. Essas iniciativas favorecem o uso inteligente da infraestrutura viária, mas estão enfrentando problemas em determinadas regiões devido a regulamentação de tais atividades. Uma alternativa e oportunidade de mercado, até então não explorada de forma automatizada, seria o melhor aproveitamento do transporte *coletivo* privado, que engloba veículos como vans, micro-ônibus, e ônibus, cujos serviços de transporte possuem uma regulamentação mais clara, e o tempo ocioso de veículos já em operação poderia ser utilizado para suprir demandas de transporte contínuo.

Neste trabalho abordamos o planejamento do transporte contínuo para requisições de um PGV e elaboramos um modelo para o Problema de Roteamento de Veículos para o Transporte Contínuo (CommuteVRP, do inglês *Commute Vehicle Routing Problem*), um problema ainda não definido na literatura, cujo objetivo é sugestão de rotas para o atendimento da demanda expressa por transporte contínuo, otimizadas com relação a fatores econômicos e mercadológicos, como lucro, número de clientes atendidos, número de veículos que prestam serviços, fidelidade, dentre outros.

Para a definição do problema, foi considerado que os serviços de transporte coletivo privado são prestados por empresas com fins lucrativos, que não trabalham de forma deficitária. Portanto, o CommuteVRP visa garantir a viabilidade econômica do conjunto de rotas sugeridas, não necessariamente atendendo a todas as requisições. O preço final cobrado de cada passageiro deve ser competitivo, nas devidas proporções de conforto e qualidade, comparado ao transporte coletivo público e ao transporte individual privado.

O CommuteVRP apresenta características dos problemas de otimização da classe de Roteamento de Veículos (*Vehicle Routing Problem*, VRP) e suas variações. Apresenta também características de um *Dial a Ride Problem* (DARP), um problema derivado dos VRP que geralmente lida com transporte de pessoas. Uma outra classe de problemas de otimização com características relevantes ao nosso trabalho é a *Team Orienteering Problem* (TOP), no qual as soluções não precisam percorrer todos os vértices e visam a maximização do lucro advindo de visitação aos vértices.

O CommuteVRP é um problema de grande relevância para a área de Sistemas de Informação, pois, trata-se de

um problema de otimização com apelo mercadológico e requer estudo e desenvolvimento de bons algoritmos capazes de resolvê-lo em tempo computacional aceitável.

2. CARACTERIZAÇÃO DO CommuteVRP

O *Commute Vehicle Routing Problem* (CommuteVRP) é muito complexo do ponto de vista computacional e depende de dados não necessariamente capturáveis no mundo real, portanto, é essencial trabalharmos em uma situação específica para encontrarmos uma proposta mercadologicamente próxima ao mundo real. Visando esta situação específica, adotamos os polos geradores de viagens (PGV) e, além disso, o CommuteVRP abordado neste trabalho é uma versão estática, na qual supomos ter os clientes e transportadores previamente cadastrados. Somando-se a isto, ainda temos as questões do custo da rota, do preço pago por passageiro, e ainda, o fato de que não queremos ter prejuízos.

O CommuteVRP visa rotas que paguem um preço justo aos transportadores, tenham um preço competitivo com rotas atrativas para os clientes e que sejam lucrativas para quem organiza e estrutura as demandas. As rotas serão criadas e ofertadas aos transportadores cadastrados, caso o transportador julgue a rota interessante, assume a rota.

O CommuteVRP tratado neste trabalho seria um CFE-VUVCCADDVRPLMW, de acordo com a classificação de Caceres-Cruz [?], com as seguintes características:

Veículos com capacidade definida **C**. Frota de veículos heterogênea **FE**, em nosso trabalho lidamos com oito tipos diferentes de veículo. Frota de veículos ilimitada **VU**. Custo variável por veículo **VC**, ou seja, cada veículo tem custo que varia baseado em um produto entre o preço por unidade de distância e a distância percorrida. Matriz de Custos assimétrica **CA**, no qual o custo de um trajeto partindo de um ponto A chegando em B pode ser diferente do custo partindo de B chegando em A. O ponto de origem dos veículos podem ser de localizações diferentes **DD**. Restrições de tamanho ou duração **L**, no caso do CommuteVRP, se aplicam tanto as restrições de tamanho quanto as de duração das viagens. Múltiplas Janelas de Tempo **MW**, neste estudo abordamos uma janela de tempo para embarque e outra para desembarque, sendo uma janela de tempo para cada passageiro. O CommuteVRP é definido em um cenário no qual temos um conjunto de pessoas que moram e trabalham ou estudam em locais diversos e possuem horários variados para chegarem ao destino. Desejamos levar essas pessoas das suas origens aos seus destinos, no horário desejado. Cada pessoa pode definir uma janela de tempo que deve embarcar e outra janela de tempo para chegar ao PGV.

O problema conta com um número indeterminado de prestadores de serviço de transporte cuja frota pode aumentar, seja pela inclusão de novos prestadores de serviços ou até mesmo pela aquisição de novos veículos por transportadores já cadastrados, caso haja rotas interessantes. Os veículos possuem capacidades diversas. Logo, estamos considerando um número ilimitado de veículos e múltiplos depósitos.

Como são vários prestadores de serviço, cada um pode ter uma localização diferente para o depósito (garagem). Além disso, como as capacidades são variadas, cada veículo tem um custo diferente por quilômetro rodado.

Como os trajetos são baseados em caminhos reais, as regulamentações de trânsito fazem com que o caminho entre dois pontos possam ser diferentes para ida e volta, consequentemente, os custos para percorrer o trajeto baseados

em distância e tempo podem ser diferentes, e assim, a matriz que os representa é assimétrica.

As rotas geradas para o CommuteVRP devem respeitar o fator de tolerabilidade afim de não comprometer o conforto dos passageiros por causa de uma rota muito longa. Por tolerabilidade entende-se que um passageiro em seu caminho real, embarcado num veículo de transporte coletivo, não pode rodar mais do que $x\%$ em quilometragem e nem gastar mais do que $y\%$ de tempo do que gastaria em seu caminho ideal, se realizasse o trajeto de carro e pelo melhor caminho entre sua origem e seu destino.

Além disso, o CommuteVRP busca otimizar o negócio e não necessariamente as rotas, ou seja, as distâncias percorridas e/ou a minimização do número de veículos envolvidos não são o principal objetivo.

O CommuteVRP não possui a obrigação de atender todos os passageiros nem de ocupar todos os veículos cadastrados, ele objetiva coletar passageiros com uma equipe de prestadores de serviço, limitado por certas restrições, visando um conjunto de rotas lucrativas.

Devido à estratégia de se otimizar o negócio, diferentemente dos problemas clássicos da literatura, os melhores resultados não estão necessariamente nas rotas mais curtas, ou mais rápidas, ou com menos veículos sendo utilizados, ou até mesmo com todos os clientes sendo atendidos.

Quando as rotas são criadas e ofertadas aos transportadores cadastrados, ainda não sabemos a localização do veículo que assumirá a rota, e por isto, o custo de se locomover da garagem até o primeiro passageiro deve ser arcado pelo transportador que se responsabilize pela rota. A rota também termina no PGV, ou seja, o trajeto de volta, do PGV à garagem (depósito), não é remunerado ao transportador.

2.1 Cômputo de Custos e Preço

As rotas geradas para o CommuteVRP devem respeitar o fator de tolerabilidade, assim como, considerar que serão executadas por transportadores que objetivam o lucro. Porém, o preço do serviço deve ser competitivo em comparação ao transporte coletivo público e em comparação ao transporte individual privado. O transporte coletivo privado (fretado) tem um nível de conforto intermediário, superior ao do transporte coletivo público mas inferior ao do transporte individual privado (taxi e similares). O preço para o passageiro deve ser compatível com este nível de conforto.

Com esta premissa e com base nos preços praticados atualmente por taxis, serviços de transporte individual de passageiros com base em aplicativos e o transporte coletivo público, em quatro grandes cidades do sudeste do Brasil, definimos que o passageiro irá pagar R\$1,00 por quilômetro percorrido e o transportador será remunerado também de acordo com os quilômetros percorridos e o tipo de veículo alocado, conforme Tabela 1.

2.2 Grafos para o CommuteVRP

O cenário para CommuteVRP é baseado no grafo da Figura 1. Um trecho de mapa com duas avenidas, sendo vias para tráfego em sentido único, e ruas perpendiculares com sentido duplo. As interseções entre as ruas são vértices, e as ruas e seus sentidos de tráfego, as arestas para um grafo direcionado. Neste caso, simulamos três requisições de clientes.

Este grafo é o que chamamos de grafo original, nele temos a estrutura que será trabalhada. A partir do grafo original,

Tabela 1: Tipos de Veículos

Nome	Capacidade	Preço por Km
CARRO	4	R\$2,15
SPIN	6	R\$2,20
VAN15	15	R\$3,00
VAN19	19	R\$3,50
MICRO23	23	R\$3,60
MICRO27	27	R\$3,80
TOCO	42	R\$4,20
TRUCKBUS	47	R\$5,40

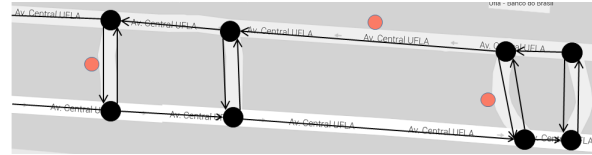


Figura 1: Grafo Original

geramos um outro grafo completo utilizando os vértices das requisições. No grafo original é calculado a menor distância entre qualquer par de requisições. O resultado deste cálculo será o peso da aresta no grafo completo de requisições.

Após a criação do grafo completo de requisições e já calculados os pesos das arestas, inserimos o vértice que representa o Polo Gerador de Viagens (PGV) que em nossa abordagem do problema será sempre o destino das requisições. Como o PGV será o destino, não é necessário arestas que saem do PGV e vão para os demais vértices, só é necessário as arestas que chegam ao PGV. A partir do grafo original é calculado o menor caminho entre o vértice de requisição até o PGV e este valor adicionado como peso das novas arestas.

Este grafo da Figura 2 representa um grafo que provavelmente possuirá uma considerável redução na quantidade do número de vértices e arestas se comparado ao grafo original, com isto, ocupando menos espaço em memória, este será o grafo trabalhado no roteamento.

2.3 Característica das Rotas

Nos problemas clássicos de Roteamento consideramos que o veículo parte de um ponto e no fim da rota deve voltar a este ponto. Neste trabalho não nos preocupamos com isto, o relevante é gerar rotas atrativas independente de onde o veículo de transporte esteja ou vá após o percurso.

Elaboramos três rotas que apresentam características do roteamento tradicional representado na Figura 3 e também três rotas para o roteamento que pretendemos, Figura 4.

Na Figura 3 temos um possível resultado de um roteamento

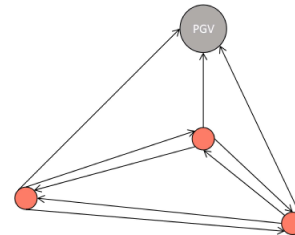


Figura 2: Grafo de requisições com PGV

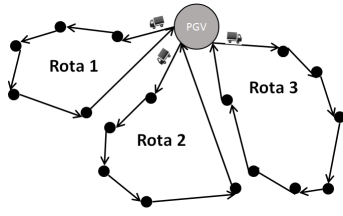


Figura 3: Roteamento tradicional

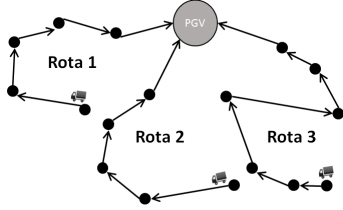


Figura 4: Roteamento pretendido

mento tradicional. Como exemplo, baseado na Rota 3, nas três primeiras requisições os passageiros estão se afastando do destino, com isto nos surge um indicativo de que o tempo que este passageiro permanece no veículo vai ser maior do que o desejado e uma possível causa de insatisfação com a rota. O mesmo acontece com as rotas 1 e 2.

Como pode ser visto na Figura 4, como não temos necessidade de fechar um ciclo, a tendência é de que a rota se inicie no ponto mais distante e a cada passageiro que embarque o veículo se aproxima do PGV.

Outro fator que sugere uma melhora na modelagem e performance da rota pretendida, ainda devido a não necessidade de um ciclo, é o fato de que as rotas pelo roteamento tradicional consideram uma aresta a mais em sua computação, o que pode afetar de forma negativa a otimização.

2.4 Modelo matemático

Esta seção apresenta algumas definições e um modelo matemático que descreve o CommuteVRP.

Seja $c(\pi)$ a função custo do caminho π , o **caminho mínimo** entre dois vértices v_i e v_j , denotado por $\pi_{ij}^\vee$, é o caminho tal que

$$c(\pi_{ij}^\vee) \leq c(\pi_{ij}), \forall \pi_{ij} \quad (1)$$

Um **trajeto** é a tupla $r(v_i, v_j, f, h)$ onde $v_i \in V$ é o vértice de origem, $v_j \in V$ é o vértice destino, $f \in \{0, 1\}$ é uma *flag* e determina se o trajeto é de ida ($f = 0$) ou de volta ($f = 1$), e h é o horário de chegada ao destino em trajetos de ida ou o horário de saída da origem em trajetos de volta.

O **preço do trajeto** $r(v_i, v_j, f, h)$ é definido com base no custo do caminho mínimo entre suas extremidades como

$$p(r) = p_b c(\pi_{ij}^\vee) \quad (2)$$

onde p_b é o preço base da unidade de custo de caminho. Como exemplo, se o custo do caminho é a distância em Km, p_b é o preço base por Km.

A **distância mínima** entre dois vértices v_i e v_j é denotada por $dist(\pi_{ij}^\vee)$.

O **tempo mínimo** entre dois vértices v_i e v_j , denotado por $time(\pi_{ij}^\vee)$ é o tempo gasto para se locomover de v_i para v_j , de carro, pelo $\pi_{ij}^\vee$.

Um caminho π_{ij} **atende** a um trajeto $r(v_s, v_t, f, h)$ quando existe um caminho π_{st} que seja um subcaminho de π_{ij} , ou seja, quando $\exists \pi_{st} \subseteq \pi_{ij}$. Quando um caminho atende a um trajeto denota-se $r(v_s, v_t, f, h) \in \pi_{ij}$ e diz-se também que o trajeto pertence ao caminho.

Define-se **rota** a relação $\rho(\pi_{ij}, R)$ entre o caminho π_{ij} e o conjunto de trajetos R tal que $\exists r(v_i, v_t, f, h) \in R, \exists r(v_s, v_j, f', h') \in R$ e, além disto $r(v_s, v_t, f, h) \in R \rightarrow r(v_s, v_t, f, h) \in \pi_{ij}$ quaisquer que sejam s e t .

Uma rota $\rho(\pi_{ij}, R)$ **atende** a um trajeto $r(v_s, v_t, f, h)$ se $r(v_s, v_t, f, h) \in R$, considerando de antemão que a definição de rota exige também que $r(v_s, v_t, f, h) \in \pi_{ij}$. Neste caso diz-se também que $r(v_s, v_t, f, h) \in \rho(\pi_{ij}, R)$, ou seja, que o trajeto r pertence à rota ρ .

Com isto podemos definir a rota $\rho(\pi_{ij}, R)$ em outras palavras como sendo um caminho π_{ij} que atende a todos os trajetos de R iniciando-se no ponto inicial de um dos trajetos de R e terminando no ponto final de um dos trajetos de R .

Um vértice $v \in V$ **pertence** a uma rota $\rho(\pi_{ij}, R)$ quando pertence ao caminho π_{ij} , ou seja, $v \in \rho(\pi_{ij}, R) \leftrightarrow v \in \pi_{ij}$.

Um vértice $v \in V$ **atende** a um trajeto $r(v_s, v_t, f, h)$ em uma rota $\rho(\pi_{ij}, R)$ se v pertence ao subcaminho de π_{ij} que atende a $r(v_s, v_t, f, h)$.

Define-se **capacidade exigida** de um vértice v em uma rota $\rho(\pi_{ij}, R)$ como sendo o número de trajetos que o vértice atende na rota, o que é o mesmo que o número de trajetos atendidos concomitantemente naquele ponto da rota.

Define-se **capacidade exigida** de uma rota $\rho(\pi_{ij}, R)$ como o máximo entre as capacidades exigidas dos vértices da rota, o que é o mesmo que o número máximo de trajetos atendidos concomitantemente na rota.

Define-se **custo de uma rota** $\rho(\pi_{ij}, R)$ para um veículo de capacidade q_v como

$$c(\rho(\pi_{ij}, R), q_v) = c_b(q_v) c(\pi_{ij}) \quad (3)$$

onde $c_b(q_v)$ é o valor base da unidade de custo de caminho para um veículo de capacidade q_v . Como exemplo, se o custo do caminho é a distância em Km, $c_b(q_v)$ é o valor base por Km para esse veículo. O custo da rota refere-se ao valor a ser pago ao Transportador, pois é o custo do que é efetivamente rodado.

Define-se a **receita de uma rota** como sendo a soma-tória dos preços dos trajetos pertencentes à rota, denota-se $rev(\rho(\pi_{ij}, R)) = \sum_{r \in R} p(r)$.

Define-se **solução** de mobilidade como um conjunto ψ de rotas tais que $\forall \rho_i(\pi_{ii'}, R_i) \in \psi, \forall \rho_j(\pi_{jj'}, R_j) \in \psi, i \neq j \rightarrow R_i \cap R_j = \emptyset$. Portanto trata-se de um conjunto de rotas tais que um trajeto não apareça em mais de uma rota.

Define-se **custo da solução** de mobilidade como

$$c(\psi) = \sum_i c(\rho_i, q_{vi}), \rho_i \in \psi \quad (4)$$

onde q_{vi} é a capacidade do veículo utilizado para executar a rota ρ_i .

Define-se **receita da solução** de mobilidade como $rev(\psi) = \sum_i rev(\rho_i), \rho_i \in \psi$. A receita da solução é o quanto os clientes pagam pelo serviço.

Define-se a **viabilidade econômica de uma rota** ρ , com base na margem de lucro operacional pré-estabelecida L_ρ , pela função booleana

$$vrou(\rho, L_\rho) = \left(1 - \frac{c(\rho)}{rev(\rho)} \geq L_\rho\right) \quad (5)$$

Se $L_\rho \rightarrow -\infty$ todas as rotas se tornam economicamente viáveis. Pode ser interessante permitir rotas economicamente inviáveis se a solução for economicamente viável, dependendo do objetivo de negócio.

Define-se a **viabilidade econômica de uma solução** ψ , com base na margem de lucro operacional pré-estabelecida L_ψ , pela função booleana

$$vsol(\psi, L_\psi) = \left(1 - \frac{c(\psi)}{rev(\psi)} \geq L_\psi\right) \quad (6)$$

Se $L_\psi \rightarrow -\infty$ todas as soluções se tornam economicamente viáveis.

Define-se **tolerabilidade de distância** $tol_d(\psi)$, com base na distância percorrida por cada passageiro pertencente à uma solução, pela função booleana

$$tol_d(\psi) = (dist(\pi_{ij}^\vee) dtol \leq dist(\pi_{ij})), \forall \rho(\pi_{ij}, R) \in \psi \quad (7)$$

Onde $dtol$ é um fator multiplicativo que corresponde à tolerabilidade (ver introdução da Seção 2) em termos de distância percorrida.

De mesmo modo, considerando $ttol$ um fator multiplicativo que corresponde à tolerabilidade (ver Seção 2) em termos de tempo de trajeto, define-se **tolerabilidade de tempo** $tol_t(\psi)$, com base na distância percorrida por cada passageiro pertencente à uma solução, pela função booleana

$$tol_t(\psi) = (time(\pi_{ij}^\vee) ttol \leq time(\pi_{ij})), \forall \rho(\pi_{ij}, R) \in \psi \quad (8)$$

Define-se a função $accept(\psi)$ como

$$accept(\psi) = tol_d(\psi) tol_t(\psi) vsol(\psi, L_\psi) (vrou(\rho, L_\rho), \forall \rho \in \psi) \quad (9)$$

e sua versão inteira como

$$iaccept(\psi) = \begin{cases} 1, & \text{se } accept(\psi), \\ 0, & \text{senão} \end{cases} \quad (10)$$

$$(11)$$

A função $iaccept(\psi)$ vale 1 se a solução for **aceitável**, ou seja, se for tolerável em termos de distância e tempo, viável, e todas as suas rotas forem viáveis.

Finalmente, define-se a função objetivo $obj(\psi)$ a ser minimizada como

$$obj(\psi) = iaccept(\psi) (c(\psi) - rev(\psi)) \quad (12)$$

Quando a solução não é aceitável, a função objetivo se iguala a zero. Se a solução for aceitável, $obj(\psi)$ terá valores negativos, se $c(\psi) < rev(\psi)$ e valores não negativos, senão. Logo, a minimização de $obj(\psi)$ é o mesmo que a maximização do lucro.

2.5 Ruin and Recreation Principle

Dentre as heurísticas e meta-heurísticas existentes na literatura para problemas semelhantes ao CommuteVRP, o *Ruin and Recreation Principle*, proposto por Schrimpf [?] tem sido repetidamente validado por outros autores, como [?], [?], [?], [?], [?] e [?], alguns destes bem recentes.

De acordo com [?], *Ruin and Recreation Principle* foi um inovador princípio de otimização, no qual, soluções primárias do problema são parcialmente, mas significativamente, arruinadas e reconstruídas ou recriadas depois disto. O método é particularmente adequado para problemas mais complexos como por exemplo, problemas descontínuos, problemas com objetivos complexos ou muitas restrições. Além

disto, para a maioria das instâncias ele alcançou os melhores resultados publicados e em várias delas conseguiu um resultado melhor ou muito melhor.

O princípio *ruin and recreation* foi implementado como única heurística para resolução de problemas do tipo caixeiro viajante e roteamento de veículos em uma ferramenta de código aberto desenvolvida em Java, denominada JSPLIT, [?]. O Jsprit é uma ferramenta muito utilizada no mercado, permite computação paralela e aceita diversas variações de problemas de roteamento de veículo incluindo vários problemas clássicos da literatura. É possível descrever o CommuteVRP de forma a ser otimizado pelo Jsprit.

Neste trabalho utilizamos o *ruin and recreation principle* e o Jsprit, fazendo as adequações necessárias. Foi alterada a função objetivo e as funções de cálculo de custo. O algoritmo trabalha com distâncias euclidianas, então foram feitas alterações para que considerasse a distância real e os tempos gastos entre trajetos fornecidos pela nossa própria matriz de custos discutida a seguir. Foram também inseridas as restrições de tolerabilidade de distância e tempo.

Uma vez que uma boa inicialização do Jsprit faz com que o algoritmo convirja mais rápido, desenvolvemos uma heurística própria para criar soluções iniciais. Ela adiciona o vértice mais distante da origem a uma rota e enquanto não ferir as restrições de tolerabilidade e capacidade do veículo, ela adiciona os vértice mais próximos ao último vértice adicionado. Caso a rota não seja lucrativa, o processo se repete desconsiderando o primeiro vértice que foi inserido na rota.

2.6 Instâncias

As instâncias para o problema foram construídas com coordenadas geográficas geradas de forma aleatória dentro de limites da área urbana de uma grande cidade brasileira. Os dados que o problema recebe como entrada são as coordenadas da localização de cada passageiro, seus respectivos destinos e janelas de tempo de embarque e desembarque. Também são passados por parâmetro os fatores de tolerabilidade de tempo e distância e as margens de lucro mínimas desejadas, por rota e para toda a solução.

É feito um pré-processamento das coordenadas geográficas das paradas para se gerar o grafo de entrada do algoritmo. Utilizando-se de ferramentas que manipulam a base de mapas do projeto *Open Street Map*, são calculadas distâncias reais e tempo de locomoção entre todos os pares de coordenadas, criando-se a matriz de custos. A matriz de custos é escrita em um arquivo conforme Figura 5. As linhas da matriz representam a origem do trajeto e a coluna o destino. Para o trajeto de A até B, podemos consultar a matriz na linha com o número referente à A e a cada duas colunas conta-se uma unidade referente à B, pois temos uma coluna pra distância e outra para tempo.

A primeira linha tem o número de coordenadas geográficas representadas no arquivo. A Figura 5 tem 8 coordenadas representadas nas duas primeiras colunas, respectivamente latitude e longitude. As seis primeiras são as requisições de embarque. Neste trabalho, consideramos que a rota se inicia no primeiro embarque, sendo assim, todos os veículos são colocados em *garagens virtuais*, cuja distância até qualquer ponto de parada deve ser zero, pois não é contabilizado no custo o deslocamento da garagem até o primeiro ponto da rota. A sétima coordenada da matriz de custos é um ponto aleatório representando a garagem virtual (GV). Note que as distâncias e tempos a partir da GV são zero, exceto nas duas

Tabela 3: 400 passageiros, 50000 iterações

T.alpha	T.Exec.(min)	R\$	N atend	mmi
0.2	4405,34	1165,557	31	5286
0.15	4208,59	1166,8095	27	5050
0.05	3481,34	1174,35476	26	4178
0	3380,48	1171,932	30	4057

Tabela 4: 300 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	433,53	787,302179	28,2	2601
10_0.15	468,88	788,75662	28	2813
10_0	213,38	787,1785	27,4	2563
10_0	218,56	788,78192	29	2623
05_0.05	210,31	785,50164	27	2522
05_0.05	208,1	788,79720	28	2497
05_0	213,38	787,1785	27,4	2563
05_0	218,56	788,78192	29	2623
01_0.05	44,14	784,41404	29,4	2648
01_0.05	43,17	787,31938	28	2590
01_0	46,42	783,69349	28	2785
01_0	42,24	787,40002	28	2534

Tabela 5: 400 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	931,5	1167,75252	31	5589
10_0.15	950,75	1161,21454	32,7	5704
05_0.05	421,5	1165,01185	30,2	5058
05_0.05	347,85	1169,93260	23	4174
5_0	431,55	1164,52517	30,5	5179
05_0	384,07	1170,33192	26	4609
01_0	90,67	1158,66345	32	5440
01_0	89,57	1165,44638	21	5374
01_0.05	74,98	1164,47790	28	4499
01_0.05	86,43	1157,90192	35,4	5186

Tabela 6: 500 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	1338,53	1669,58434	34	8031
10_0.15	1456,25	1662,85091	33	8737
10_0	1204,16	1671,95782	30	7225
10_0	1361,39	1666,72535	34,1	8168
01_0.05	143,69	1664,17110	35	8621
01_0.05	142,87	1655,63722	31,8	8572
01_0	139,35	1664,90044	27	8361
01_0	139,61	1655,88095	31,2	8377

Tabela 7: 600 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	7555,15	2015,48404	22	45331
10_0.15	7515,55	2003,91150	26,4	45093
01_0.05	615,68	2004,69496	40	36941
01_0.05	681,67	1995,04722	32,6	40900
01_0	477,69	2011,83050	24	28661
01_0	529,65	1994,13455	28,2	31779

Tabela 8: 700 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	5358,11	2399.00636	44	32149
01_0	455,03	2408,99016	42	27302
01_0	452,69	2398,92099	43,9	27161

Tabela 9: 800 passageiros

Exec.	T. Exec(min)	R\$	N atend	mmi
10_0.15	6330,16	2833,52600	43	37981
01_0	548,8	2810,68164	39	32928
01_0	498,28	2802,40381	38,3	29897

3.2 Discussão sobre os resultados

Para 300 e 400 passageiros foram realizados testes com 50 mil iterações cada execução, variando apenas o threshold_alpha. Podemos observar nas Tabelas 2 e 3 que os threshold_alpha setados em 0.05 e 0 apresentaram os melhores resultados, tanto em lucro quanto os que executaram em menor tempo. Destaque para o threshold_alpha 0.05 que obteve maior lucro nos dois casos e execução mais rápida no primeiro caso. Os demais testes com 300 e 400 passageiros confirmaram esta tendência.

À medida que executamos as instâncias com um número superior de iterações, para um mesmo threshold_alpha, o lucro obtido para esta instância sempre aumenta, porém, não de forma tão significativa quando comparado ao tempo extra gasto na execução. De acordo com o threshold_alpha setado, o algoritmo aceita soluções piores por um certo tempo que obedece uma função baseada também no número de iterações, ou seja, quanto maior o número de iterações, mais iterações são gastas aceitando soluções piores e dificultando a convergência do algoritmo. Em contrapartida, se não aceitarmos soluções piores, podemos mais facilmente ficar presos em mínimos locais. Se o número de iterações for muito pequeno, o algoritmo pode não ter tempo suficiente para achar soluções melhores. Outra característica do algoritmo é ter uma iteração mais rápida à medida que a solução melhora. Dentro de uma iteração ele testa várias possíveis soluções e quando a solução é boa, enquanto constrói uma nova solução, já pode detectar que será pior do que a atual e eliminá-la, ou seja, o tempo gasto por iteração converge para um tempo mínimo de processamento por iteração.

Com os testes em instâncias grandes com 500, 600, 700 e 800 passageiros, foi possível observar que o algoritmo foi capaz de encontrar soluções lucrativas em todos os cenários apontados.

4. CONCLUSÃO

O objetivo do trabalho foi estudar o Problema de Roteamento de Veículos para o Transporte Contínuo (CommuteVRP), apresentar uma modelagem matemática e encontrar soluções otimizadas para o problema visando aspectos mercadológicos.

O CommuteVRP foi estruturado em um problema de roteamento de veículos com diversas características, como por exemplo, veículos capacitados, frota ilimitada e heterogênea; com oito tipos diferentes de veículos com capacidades variadas e cada veículo possuindo um custo diferente por unidade de distância percorrida, múltiplas janelas de tempo, coleta e entrega de passageiros, múltiplas origens para as rotas, e

restrições de tempo e distância para cada passageiro.

Diferente dos problemas clássicos da literatura, as rotas não são um ciclo, começam no embarque do primeiro passageiro e terminam no PGV. Além disto, o objetivo é maximizar o lucro levando em conta que cada rota individualmente deve ter uma margem mínima de lucro e a solução como um todo (todas as rotas somadas), uma outra margem de lucro. Com esses objetivos, nem todos os passageiros precisam ser atendidos e nem todos os transportadores alocados. A distância percorrida pelo transportadores não é necessariamente a mínima, assim como o número de veículos utilizados não deve ser minimizado, apesar de que empiricamente este aspecto tenda a ser considerado.

As instâncias abordadas neste trabalho são relativamente grandes e tem características que buscam emular situações reais. Este conjunto de características, até então, não haviam sido abordadas simultaneamente na literatura, sendo assim, este trabalho conseguiu prover rotas lucrativas para um problema de alta complexidade, sendo uma contribuição na área computacional ao instigar que trabalhos futuros sejam realizados afim de encontrar melhores resultados para as instâncias utilizadas, ou ainda que, é possível encontrar soluções para problemas que combinam variantes do problema de roteamento de veículos para instâncias grandes que se assemelham a problemas práticos do mundo real. Foi possível observar que a heurística do princípio *ruin and recreation* aplica-se de forma satisfatória ao CommuteVRP e é capaz de encontrar soluções lucrativas em tempos aceitáveis de computação.

Outra contribuição do trabalho é que ele pode servir de base para a definição de modelos de negócio para soluções que lidam com o transporte sob demanda (aplicativos e transporte coletivo privado). O CommuteVRP favorece o compartilhamento de veículos e é uma alternativa sustentável para prover mobilidade utilizando-se da infraestrutura já existente. Com isto, eventualmente atenuando os conhecidos problemas de mobilidade urbana e reduzindo a poluição.

O trabalho pode ser aplicado a empresas e instituições que queiram prover transporte aos funcionários, ou prefeituras e concessionárias do transporte público na criação de novas linhas.

Podemos considerar que os objetivos foram alcançados, pois foram obtidas rotas lucrativas para todas as instâncias experimentadas. Ainda existe muito trabalho a ser realizado nesta área e com o conhecimento adquirido, expandimos o entendimento sobre o que ainda pode ser explorado.

4.1 Trabalhos futuros

Como trabalhos futuros, podemos abordar alguns pontos tais quais:

1. Elaborar formas diferentes do computo de preços, tanto para os valores pagos pelo passageiro quanto aos valores pagos aos transportadores;
2. Determinar limitantes para as instâncias afim de realizar uma análise qualitativa dos resultados;
3. Implementar e adequar outras propostas de abordagem para lidar com problemas semelhantes ao CommuteVRP;
4. Desenvolver abordagens específicas para o problema;

5. Lidar com fidelização de passageiros ou transportadores, que no início do processamento são determinados a fazerem parte da solução;
6. Maximizar o número de clientes atendidos, desde que as rotas tenham um custo determinado, pensando na publicidade em atrair novos clientes;
7. Maximizar o número de veículos utilizados, desde que as rotas tenham um custo determinado, priorizando aumentar o cadastro na base de dados de transportadores;
8. E realizar estudos com instâncias ainda maiores.

5. AGRADECIMENTOS

Os autores agradecem pelo suporte obtido da Capes, CNPq e Fapemig.

6. REFERÊNCIAS

- [1] R. Bent and P. Van Hentenryck. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4):515–530, 2004. cited By 138.
- [2] O. Bräysy, W. Dullaert, G. Hasle, D. Mester, and M. Gendreau. An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle-routing problem with time windows. *Transportation Science*, 42(3):371–386, 2008. cited By 38.
- [3] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan. Rich vehicle routing problem: Survey. *ACM Comput. Surv.*, 47(2):32:1–32:28, Dec. 2014.
- [4] M. Dell’amico, M. Monaci, C. Pagani, and D. Vigo. Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. *Transportation Science*, 41(4):516–526, 2007. cited By 28.
- [5] DESA/UN. World’s population increasingly urban with more than half living in urban areas, 2014.
- [6] A. Kovacs, B. Golden, R. Hartl, and S. Parragh. The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816, 2015. cited By 4.
- [7] R. Masson, F. Lehuédé, and O. Péton. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355, 2013. cited By 18.
- [8] J. M. Rodrigues. Crescimento da frota de automóveis e motocicletas nas metrópoles brasileiras 2001/2011. Relatório Técnico 10/2012, PROURB/UFRJ, Rio de Janeiro, 2012.
- [9] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006. cited By 325.
- [10] R. S. SCARINGELLA. A crise da mobilidade urbana em são paulo, 2001.
- [11] S. Schröder. Jsprit, 2016.
- [12] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139 – 171, 2000.

indicate the start of each Appendix, with alphabetic order