O PROBLEMA DO CAIXEIRO VIAJANTE COM MÚLTIPLOS PASSAGEIROS E QUOTA

Allan Vilar de Carvalho
Universidade Federal do Rio Grande do Norte
Campus Universitário, Lagoa Nova, Natal, RN
allanvilarcarvalho@gmail.com

Marco C. Goldbarg
Universidade Federal do Rio Grande do Norte
Campus Universitário, Lagoa Nova, Natal, RN
marcocgold@gmail.com

Elizabeth F. G. Goldbarg
Universidade Federal do Rio Grande do Norte
Campus Universitário, Lagoa Nova, Natal, RN
beth@dimap.ufrn.br

RESUMO

Este trabalho apresenta o Problema do Caixeiro Viajante com Múltiplos Passageiros e Quota, uma variante do Problema do Caixeiro Viajante com Quota relacionada com outros modelos de roteamento enriquecido. O problema consiste em desenvolver uma solução do problema do Caixeiro Viajante com Quota considerando que o caixeiro é o motorista de carro que deseja compartilhar os custos de viagem com passageiros que podem ser embarcados e desembarcados em cidades da sua rota. Cada passageiro cuja solicitação de transporte é atendida compartilha os custos da viagem no caminho entre sua origem e seu destino com o caixeiro e outros passageiros no carro. Um modelo de programação matemática e um banco de instâncias são propostos para o problema. As soluções exatas para instâncias com até 10 vértices são apresentadas. Uma heurística ad hoc e duas heurísticas baseadas na meta-heurística colônia de formigas são propostas. Os resultados de um experimento computacional são relatados.

PALAVRAS-CHAVE. Caixeiro Viajante. Problemas de Ridesharing. Meta-heurísticas.

Tópicos (MH – Meta-heurísticas. OT – Otimização)

ABSTRACT

This study presents the Traveling Salesman with Multiple Passengers and Quota Problem, a variant of the Traveling Salesman Problem with Quota that is also related to other enriched routing models. The problem consists in solving the Traveling Salesman with Quota considering that the salesman is the driver of a car who wants to share travel costs with passengers that can be picked up and droped-off in the cities of his route. Each passenger whose transport request is satisfied shares the travel costs of the path from his origin to his destination with the salesman and other passengers in the car. A mathematical programming model and a set of instances are proposed for the problem. Optimal solutions for instances up to 10 vertices are presented. An ad hoc heuristic and two heuristics based on the ant colony metaheuristic are proposed. The results of a computational experiment are reported.

KEYWORDS. Travelling Salesman. Ridesharing Problems. Metaheuristics.

Paper topics (MH – Metaheuristics. OT – Optimization)

https://proceedings.science/p/85535?lang=pt-br

1. Introdução

O Problema do Caixeiro Viajante (PCV) é um dos problemas de Otimização Combinatória mais notáveis da literatura. Uma importante variante do PCV é o Caixeiro Viajante com Coleta de Prêmios [Balas 1989], onde o caixeiro visita as localidades de uma rede, com objetivo de minimizar seus custos de viagem e penalidades associadas, sendo ainda obrigado a coletar um valor mínimo de bônus que é distribuído nas localidades. Nesse caso o caixeiro está desobrigado de visitar todas as localidades de sua rede de visitas. Awerbuch et al. [1998] introduzem uma variante nesse problema considerando apenas os custos e os bônus, e o denominam de Caixeiro Viajante com Quota (PCVQ). Outras variantes são relatadas na literatura. Uma estruturação sistemática dos problemas de coleta de prêmios é encontrada em Goldbarg e Goldbarg [2012] que propõem uma classificação dos problemas dessa classe em cinco diferentes grupos.

O problema objeto da presente pesquisa é denominado de Problema do Caixeiro Viajante com Múltiplos Passageiros e Quota (PCV-MPQ) constituindo-se em uma variante do PCVQ onde, além do caixeiro definir uma rota de mínimo custo que obrigatoriamente coleta uma quota mínima de bônus distribuídos sobre os vértices, o caixeiro pode compartilhar acentos em seu carro com um conjunto de passageiros distribuídos nas localidades da rede. Ao fazer isso, pode ratear uniformemente os custos de cada aresta da rota com os passageiros que ocupam o carro ao atravessar a aresta.

O caixeiro pode embarcar um passageiro caso visite a sua localidade de embarque. Caso um passageiro seja embarcado, o caixeiro obrigatoriamente deverá visitar a localidade de desembarque desse passageiro. Cada passageiro possui um limite máximo de despesas para realizar o trajeto entre a sua localidade de embarque e a de desembarque, independentemente da rota escolhida pelo caixeiro. Os custos de um passageiro embarcado correspondem à soma da parcela de rateio que lhe couber em função da ocupação do carro, em cada aresta percorrida pelo carro mantendo esse passageiro embarcado. Os custos são rateados de forma uniforme entre os passageiros embarcados mais o caixeiro.

No caso do PCV-MPQ, a seleção dos vértices e sua ordem de visita estão acopladas com a decisão de coleta de bônus e de embarque e desembarque de passageiros. A contribuição do embarque de passageiros em um vértice não pode ser reduzida a prêmios previamente conhecidos ou a penalidades fixas, uma vez que seu valor depende da permanência do passageiro no carro. A permanência no carro depende dos vértices selecionados, da sequência de visitas realizada pelo caixeiro e da capacidade do carro. Os custos do passageiro embarcado são obtidos da soma do rateio uniforme entre os passageiros embarcados e o caixeiro em cada aresta percorrida pelo passageiro.

Considerando o objetivo principal desse modelo de roteamento enriquecido, são também visíveis elementos em comum com os modelos denominados de *Pick up and Delivery* [Parragh et al. 2008] e, em especial, com a classe dos problemas denominados de *Ridesharing* [Amey et al. 2011].

O atual trabalho é organizado da seguinte forma. A seção 2 contextualiza o problema proposto em relação ao estado da arte. A seção 3 apresenta a formulação matemática do PCV-MPQ. A seção 4 descreve a formação das instâncias de teste do problema. A seção 5 apresenta a experimentação da solução exata da formulação proposta. A seção 6 descreve um algoritmo heurístico *ad hoc*. A seção 7 apresenta dois algoritmos em colônia de formiga para a aproximação do problema. A seção 8 realiza um experimento computacional comparativo para avaliar o potencial dos algoritmos propostos nas seções 5 e 6. A conclusão do trabalho é apresentada na seção 9.

2. Contextualização do PCV-MPQ no estado da arte

Entende-se por transporte em grupo ou transporte solidário, o uso compartilhado de um veículo por diferentes pessoas. De um modo geral, qualquer sistema de transporte coletivo pode ser considerado uma forma genérica de transporte solidário. Contudo, o transporte dito solidário é

mais restritamente definido na literatura, envolvendo prioritariamente o compartilhamento de veículos de propriedade privada [Nourinejad 2014]. Essa modalidade de transporte possui potencial para, principalmente, aumentar a taxa de ocupação dos veículos particulares. Esse fato pode reduzir os custos de transporte urbano, minimizar impactos ambientais de poluição, diminuir tempos de viagens, engarrafamentos e reduzir áreas nobres destinada ao estacionamento de carros. Finalmente, tem potencial para melhorar o funcionamento dos sistemas públicos de transporte, por agregar a opção de trajetos complementares, bem como aumentar o conforto e segurança do usuário. O tema do transporte em grupo ou solidário é crescentemente explorado na literatura. A prática é incentivada em países como os Estados Unidos, Canadá e alguns estados da União Europeia. O transporte solidário pode ser realizado segundo diversas arquiteturas. Alguns exemplos desses compartilhamentos são os denominados *bike-sharing* (uso coletivo de bicicletas, normalmente em trajeto que faz a ligação de terminais de transporte), *carsharing* (carros alugados por tempo definido), *ridesharing* (compartilhamento de acentos nos carros), *personal vehicle sharing* (carro pessoal que pode ser alugado), *carpooling* (lotação de veículos), *slugging* (carona concedida em filas localizadas nas proximidades do acesso às pistas livres de pedágio e que exigem carros com vários ocupantes) e *taxi-sharing* (compartilhamento de viagens de taxi) [Nourinejad 2014].

2.1. O PCV-MPQ e suas ligações com outros modelos de roteamento enriquecido

Através da formulação matemática do PCV-MPQ que será apresentada no item 3, é tipicamente visível sua conexão com o Problema do Caixeiro Viajante com Quota, segundo a formulação proposta por Awerbuch et al. [1998]. O modelo do PCV com Quota é pouco visitado na literatura, destacando-se os trabalhos de Ausiello et al. [2004] que relatam um algoritmo para o *On-Line Quota Traveling Salesman Problem*. Yu et al. [2014] relatam algoritmos exatos para a variante *online* desse problema.

Ainda são visíveis possíveis conexões com o problema de roteamento do tipo *Pick up and Delivery* em virtude da característica do embarque e desembarque de passageiros ser considerada em algumas variantes dessa classe [Doerner e Salazar-González 2014]. Todavia, o PCV-MPQ não se obriga a atender uma demanda de passageiros e, nesse sentido, não pertence às clássicas variantes de *Pick up and Delivery* [Parragh et al. 2008].

2.2. O PCV-MPQ e suas ligações com o *Ridesharing*

O PCV-MPQ define uma forma peculiar de transporte solidário oportunista. O caixeiro viajante programa sua rota em função exclusiva de seu interesse e desconhece qualquer imposição de atendimento da demanda dos passageiros. Por outro lado, o caixeiro possui a necessidade de cumprir objetivos próprios de visita das localidades, modelados através da coleta de um bônus mínimo.

Os problemas de *Ridesharing* são definidos de várias formas na literatura [Amey et al. 2011]. De uma forma geral, corresponde ao problema de reunir duas ou mais pessoas para partilhar uma única viagem em um veículo, sem levar em conta um acordo prévio ou um histórico de cooperação [Dailey et al. 1999]. Nesse sentido geral, o PCV-MPQ poderia ser classificado como um problema de *Ridersharing*. Todavia o *Ridesharing*, na literatura presente, é omisso nas situações típicas de embarque e desembarque mais intenso. Classicamente é um modelo de lotação de carros para uma dada viagem previamente definida pelo motorista. Mas em ampla visão o *Ridesharing* está se mostrando cada vez mais flexível, já considerando englobar um serviço de taxi ou de transporte personalizado, todavia seguindo um trajeto e condições definidas previamente [Chan e Shaheen 2012]. A importância do aspecto ocasional e oportunista da reunião de motoristas e passageiros ganha cada vez mais espaço, como é ressaltada em vários trabalhos na literatura de forma que as versões “real-time” [Gruebele 2008] ou “dinâmica” [Deakin et al. 2010] do *Ridesharing* ocupam grande atenção do esforço de pesquisa atual. Outras implicações são possíveis além da lotação do carro, com impactos em áreas como transportes, economia e ciências sociais

[Kleiner et al. 2011].

Ridesharing pode ser visto como relacionado ao *Carpooling* e ao *dial-a-ride*. O *Carpooling* diferencia-se por tratar de demanda regular, com destinos pouco variáveis para os participantes e com longo horizonte de planejamento [Morency 2007]. Por outro lado, o problema também guarda elementos de semelhança com o denominado *dial-a-ride* que busca atender viagens compartilhadas a partir de pedidos de origem e destino variados em uma área específica, voltado para um público alvo quase sempre bem conhecido e, usualmente, atendidos por um conjunto de carros projetados especificamente para o transporte dessa demanda [Berbeglia et al. 2010].

Conclui-se que o PCV-MPQ compartilha elementos em comum com vários modelos de roteamento enriquecido e *Rideshering*, todavia formando uma aplicação com características peculiares próprias.

3. Formulação matemática do PCV-MPQ

O PCV-MPQ é definido sobre um grafo $G = (N, M, B, P)$, onde $N = \{1, ..., n\}$ é o conjunto de vértices que representam as cidades, $M = \{1, ..., m\}$ é conjunto de arestas que representam as estradas que ligam as cidades e $B = \{b_1, ..., b_n\}$ é conjunto de bônus associado à cada cidade, considerando-se $b_s=0$, onde s é a cidade em que o caixeiro inicia e finaliza seu percurso. O bônus é coletado no momento da visita. $P = \{1, ..., \#p\}$ é o conjunto de passageiros. A cada $p \in P$, estão associados uma origem, o_p, um destino, d_p, $o_p \neq d_p$, e uma quantia máxima, w_p, que p está disposto a pagar para realizar sua viagem no carro do caixeiro. A cada aresta $(i, j) \in M$ está associado um custo, c_{ij}, que é dividido igualmente com todos os ocupantes do carro e representa o custo do deslocamento entre as cidades i e j. Na formulação do PCV-MPQ, apresentada nas expressões (1)-(18), x_{ij} é uma variável binária com valor será 1 se a aresta (i, j) é utilizada e 0, caso contrário, v_{ij}^p, também é uma variável binária que terá valor 1 se o passageiro p é transportado pela aresta (i,j), 0 caso contrário. A quota mínima é representada por Q.

$$\text{Minimizar } Z = \sum_{(i,j)\in M} \frac{x_{ij}c_{ij}}{\sum_{p\in P} v_{ij}^p + 1} \tag{1}$$

Sujeito a:

$$\sum_{i\in N\setminus\{j\}} x_{ij} \leq 1 \quad \forall j \in N\setminus\{s\} \tag{2}$$

$$\sum_{i\in N\setminus\{j\}} x_{ji} \leq 1 \quad \forall j \in N\setminus\{s\} \tag{3}$$

$$\sum_{i\in N\setminus\{s\}} x_{si} = 1 \tag{4}$$

$$\sum_{i\in N\setminus\{s\}} x_{is} = 1 \tag{5}$$

$$\sum_{i\in N\setminus\{j\}} x_{ij} - \sum_{i\in N\setminus\{j\}} x_{ji} = 0 \quad \forall j \in N\setminus\{s\} \tag{6}$$

$$u_i - u_j + (n-1)x_{ij} \leq n-2 \quad 2 \leq i, j \leq n,\ i \neq j \tag{7}$$

$$\sum_{\substack{(i,j)\in M \\ i\neq j}} b_i x_{ij} \geq Q \tag{8}$$

$$\sum_{p\in P} v_{ij}^p \leq K x_{ij} \quad \forall (i, j) \in M,\ i \neq j \tag{9}$$

$$\sum_{(i,j)\in M} \frac{v_{ij}^p c_{ij}}{\sum_{p\in P} v_{ij}^p + 1} - w_p \leq 0 \quad \forall p \in P,\ i \neq j \tag{10}$$

$$\sum_{j\in N\setminus\{i\}} v_{ji}^p - \sum_{j\in N\setminus\{i\}} v_{ij}^p = 0 \quad \forall p \in P,\ i \in N\setminus\{o_p,\ d_p\} \tag{11}$$

$$\sum_{i\in N\setminus\{o_p\}} v_{io_p}^p = 0 \quad \forall p \in P \tag{12}$$

$$\sum_{i\in N\setminus\{d_p\}} v_{d_p i}^p = 0 \quad \forall p \in P \tag{13}$$

$$\sum_{i\in N\setminus\{s\}} v_{si}^p = 0 \quad \forall p \in P,\ o_p \neq s \tag{14}$$

$$1 \leq u_i \leq n-1 \quad 2 \leq i \leq n \tag{15}$$

$$v_{ij}^p \in \{0,1\} \quad \forall (i,j) \in M,\ \forall p \in P \tag{16}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in M \tag{17}$$

$$c_{ij} \in \mathbb{R}^+ \quad \forall (i,j) \in M \tag{18}$$

A função objetivo representada em (1) visa minimizar o custo total do caixeiro. Seu cálculo é realizado em rateio igualitário entre os ocupantes do carro, considerando o caixeiro como um dos ocupantes. As restrições (2) e (3) estabelecem as condições de formação de uma única rota considerando um subconjunto de localidades de *N*. As restrições (4) e (5) garantem o início e o término da rota no vértice origem. A restrição (6) obriga que qualquer vértice visitado pelo caixeiro possua uma aresta de entrada e uma de saída. A restrição (7) garante a formação de um único ciclo na solução [Miller et al. 1960]. A restrição (8) obriga a coleta da quota mínima de bônus *Q*. A restrição (9) garante que a capacidade máxima do carro, *K*, não será ultrapassada. A restrição (10) proporciona que a despesa total de cada passageiro $p \in P$ não ultrapasse w_p. A restrição (11) garante que os passageiros embarcados serão somente desembarcados no seu destino. As restrições (12) e (13) garantem que os passageiros não são embarcados antes do vértice de embarque ou que permaneçam no carro após o caixeiro visitar o vértice de destino do passageiro. A restrição (14) garante que um passageiro somente poderá ter como destino a cidade inicial caso não tenha sido embarcado na cidade inicial. As restrições (15) a (18) estabelecem os limites e a natureza das variáveis de decisão.

4. Formação das instâncias de teste do PCV-MPQ

Até onde é do conhecimento dos autores, o PCV-MPQ não foi ainda examinado na literatura. Portanto, o trabalho propõe um conjunto de instâncias para auxiliar na validação de algoritmos de solução. São construídos três grupos de instâncias, com matriz de distância simétrica e assimétrica e possuindo diferentes características visando reduzir a introdução de viés. A distribuição numérica das instâncias é projetada para reduzir a formação de viés no banco de casos teste. Os custos das arestas e o valor do bônus dos vértices são definidos aleatoriamente em sorteio uniforme. Os custos das arestas variam entre 100 e 250. Nos casos simétricos, o valor é sorteado para a aresta. Nos casos assimétricos para cada direção da aresta. O valor do bônus dos vértices varia entre 0 e 250. A capacidade do carro é de 3, 4 e 5 passageiros. A quantidade de passageiros por vértice varia entre 0 até uma ou duas vezes a capacidade do carro. O destino de cada passageiro é sorteado aleatoriamente. Nas instâncias do grupo 2 e 3, por sorteio aleatório e equiprovável da instância, selecionam-se 10%, 15% ou 20% dos vértices da instância para que as arestas e os bônus sejam redefinidos. Os custos das arestas das instâncias sorteadas no grupo 2 variam entre 400 e 700. No grupo 3 esses custos variam entre 250 e 400. Os bônus nas instâncias sorteados variam entre 300 e 500. As instâncias são nomeadas por quatro atributos básicos de formação separados por um traço, a saber: número de localidades – quantidade de passageiros na instância – capacidade do carro – grupo. No experimento de validação são usadas doze instâncias em cada um dos três grupos, conforme resume a tabela 1. No processo de parametrização foram utilizadas 24 outras instâncias, obtidas pelo mesmo processo de formação e com parâmetros definidos aleatoriamente, 8 localizadas em cada grupo.

5. Solução exata da formulação do PCV-MPQ

A solução exata do modelo proposto no item 3 foi implementada no software Gurobi versão 7.5.1. Todos os testes computacionais foram executados em um computador com processador Intel Core i7-2600K de 3.40 GHz, 4 GB de RAM e sistema operacional Windows 7. Os algoritmos foram programados em C++. Na tabela 1, a coluna (Instância) relaciona as características da instância, a coluna (C) apresenta o valor obtido na solução e a coluna (T) o tempo necessário para a execução do algoritmo, em segundos. Os asteriscos sinalizam as instâncias que possuem limite máximo do número de passageiros nas localidades igual a duas vezes a capacidade do carro. As instâncias sem a marcação de asterisco, possuem esse limite igual à capacidade do carro. Foi dado ao solver um limite de tempo de $8x10^4$ segundos. As instâncias para as quais a solução ótima foi encontrada pelo solver, possuem tempo associado na coluna *Exato*, menor que $8x10^4$.

Como se observa na tabela 1, dentro do horizonte de tempo de $8x10^4$ segundos, 27 das 36 instâncias não puderam ser solucionadas de forma exata pelo solver.

Tabela 1 – Resultados do algoritmo exato e da heurística Ah

Instância	Exato		Heurística *Ah*		Instância	Exato		Heurística *Ah*	
	C	T	C	T		C	T	C	T
10-18-4-1	207,55	1244,93	475,33	0,09	10-21-4-1	199,50	2493,08	296,83	0,07
10-34-5-1	145,32	$8x10^4$	227,00	0,06	10-49-5-1*	150,33	$8x10^4$	226,08	0,07
20-26-3-1	340,83	$8x10^4$	433,83	0,06	20-26-3-1	361,58	$8x10^4$	374,67	0,07
20-114-5-1*	759,22	$8x10^4$	238,58	0,07	20-95-5-1*	357,20	$8x10^4$	206,27	0,09
30-101-3-1*	2077,00	$8x10^4$	492,17	0,06	30-89-3-1*	2252,00	$8x10^4$	405,67	0,08
30-122-4-1*	2920,00	$8x10^4$	392,45	0,08	30-56-4-1	2389,00	$8x10^4$	881,50	0,06
10-14-3-2	338,33	1477,78	767,00	0,05	10-17-3-2	329,50	3626,44	893,00	0,05
10-18-4-2	344,00	14721,30	935,00	0,05	10-21-4-2	398,45	19308,30	544,83	0,05
20-26-3-2	677,42	$8x10^4$	1965,00	0,05	20-26-3-2	431,00	$8x10^4$	847,50	0,07
20-114-5-2*	729,97	$8x10^4$	656,00	0,06	20-95-5-2*	498,28	$8x10^4$	1809,00	0,07
30-122-4-2*	3542,00	$8x10^4$	1394,67	0,07	30-105-4-2*	2757,00	$8x10^4$	512,90	0,08
30-155-5-2*	3285,00	$8x10^4$	942,67	0,07	30-143-5-2*	4617,00	$8x10^4$	1010,17	0,07
10-14-3-3	292,75	2233,11	622,00	0,05	10-17-3-3	294,42	4600,82	647,00	0,05
10-34-5-3	154,83	$8x10^4$	259,75	0,06	10-24-5-3	189,30	6887,94	490,00	0,05
20-47-4-3	554,28	$8x10^4$	709,67	0,06	20-42-4-3	454,70	$8x10^4$	523,41	0,06
20-114-5-3*	982,50	$8x10^4$	390,33	0,17	20-38-5-3	470,98	$8x10^4$	1035,00	0,04
30-35-3-3	2876,00	$8x10^4$	2087,00	0,05	30-50-3-3	2432,50	$8x10^4$	1738,00	0,07
30-155-5-3*	2251,00	$8x10^4$	930,33	0,06	30-143-5-3*	2939,00	$8x10^4$	930,33	0,06

6. Solução aproximada para o PCV-MPQ com o uso da Heurística *ad hoc* (Ah)

Visando obter soluções viáveis e examinar o potencial de compor estratégias com buscas locais nas etapas de decisão do problema a saber: rota, escolha dos vértices de visita e esquema de embarque de passageiros, a pesquisa desenvolveu uma heurística *ad hoc* que seleciona vértices de visita de forma a satisfazer a quota mínima para, posteriormente e de forma sequencial, encontrar uma rota barata que una os vértices selecionados e, a seguir, determine um esquema de embarque ótimo. Para a determinação da rota, a heurística desenvolve uma busca local sobre os vértices selecionados com base na heurística Lin-Kernighan [Lin e Kernighan 1973].

Ah ()
1. R ← conjunto de vértices;
2. R' ← busca Lin-Kernighan (R);
3. Carregamento-otimo(R');
4. Retorne R';

Quadro 1 – Algoritmo *Ah*

O embarque do carro é, então, realizado de forma exata para a rota encontrada usando o modelo de programação matemática apresentado no trabalho de Calheiros [2017], fixando as variáveis de rota, com o auxílio do solver Gurobi. A heurística *Ah* foi implementada e executada na mesma plataforma dos testes computacionais do modelo de solução exata.

Da análise dos resultados comparativos da tabela 1, observa-se que os valores obtidos pelo algoritmo *Ah* quando o algoritmo exato encontra a solução ótima, estão afastados, no melhor

caso, cerca de 40% do valor da solução ótima encontrada para a instância. Para o caso das instâncias simétricas 20-114-5-1*, 30-101-3-1*, 30-122-4-1*, 20-114-5-2*, 30-122-4-2*, 30-155-5-2*, 20-114-5-3*, 30-35-3-3 e 30-155-5-3*, bem como para as assimétricas 20-95-5-1*, 30-89-3-1*, 30-56-4-1, 30-105-4-2*, 30-143-5-2*, 30-50-3-3 e 30-143-5-3*, o algoritmo *Ah* alcança melhores valores que o procedimento exato interrompido no tempo. A heurística *ad hoc* obtém soluções para todas as instâncias do teste.

7. Algoritmos em Colônia de Formigas para a solução do PCV-MPQ

O presente item relata dois algoritmos meta-heurísticos inspirados no Ant Colony System (ACO) diferenciando-se por relatar um algoritmo clássico em colônia de formigas (ACO) e um algoritmo em colônia de formigas com refinamento através de um procedimento de busca local (ACO-VNDP). Os algoritmos são inspirados na proposta de Dorigo e Di Caro [1999], com as adaptações necessárias ao contexto das várias decisões do problema.

No modelo básico ACO as formigas coordenam a atratividade das rotas desenvolvidas através de 2 parâmetros definidos por feromônio τ e visibilidade η. Como regra geral, os caminhos mais visitados acabam recebendo o depósito de mais feromônio e tornando-se mais atrativos para a escolha das formigas. A fórmula que modula o processo é expressa na equação (20), onde N_i^k representa as localidades vizinhas da localidade *i* que ainda não foram visitadas pela formiga k. α e β são parâmetros que controlam o peso relativo do feromônio e da visibilidade dada à aresta. Um processo de atualização do feromônio é realizado a cada ciclo do algoritmo e pode ser feito de 3 maneiras, como abordado por Dorigo e Di Caro [1999]. A maneira Ant-cycle foi a escolhida e utiliza a equação (21), onde $\rho \in [0, 1]$ representa o processo de evaporação do feromônio e $\Delta\tau$ é o cálculo do novo depósito de feromônio representado em (22). $f_k(\vec{x})$ representa o custo da rota.

$$p_{ij}^k = \begin{cases} \frac{\tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{x \in N_i^k} \tau_{i,x}^{\alpha} \cdot \eta_{i,x}^{\beta}} & \forall j \in N_i^k \\ 0 & caso\ contrário \end{cases} \quad (20)$$

$$\tau_{i,j} = (1 - p) \cdot \tau_{i,j} + \Delta_{\tau} \quad (21)$$

$$\Delta_{\tau} = \frac{1}{f_k(\vec{x})} \quad (22)$$

Considerando as peculiaridades do PCV-MPQ o presente trabalho propõe uma nova formula para o cálculo da probabilidade de escolha de uma cidade $j \in N_i^k$, o conjunto de localidades vizinhas à localidade *i* e não visitada pela formiga k. A fórmula visa considerar na modulação de depósito de feromônios e visibilidade a atratividade do bônus associados às localidades e o potencial de embarque de passageiros. A equação (23) apresenta a fórmula considerada, onde α, β e γ representam o peso relativo do feromônio, peso relativo do custo da aresta e o peso relativo do bônus do vértice, respectivamente, λ_j denota a quantidade de bônus no vértice j, $\eta_{ij} = \frac{1}{c_{ij}}$ denota a atratividade da aresta (i,j) e ω_j denota a quantidade de passageiros com origem no vértice *j* e destino ainda não visitado pela formiga *k*.

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta} \cdot [\lambda_j]^{\gamma} \cdot \omega_j}{\sum_{j \in N_i^k} [\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta} \cdot [\lambda_j]^{\gamma} \cdot \omega_j} & \forall j \in N_i^k \\ 0 & caso\ contrário \end{cases} \quad (23)$$

O algoritmo ACO desenvolvido é apresentado no quadro 2. O ACO realiza no passo 1 a definição da variável NC que controla o número máximo de ciclos NC_{max}. O passo 2 inicializa o feromônio e visibilidade de todas as arestas. O passo 3 inicializa um laço que é executado NC_{max}

vezes, sendo que a cada execução, o passo 4 define a posição inicial das #*m* formigas. O passo 5 executa um laço que constrói #*m* formigas utilizando a probabilidade segundo a equação (23) para selecionar a cidade j a adicionar na formiga k. O passo 9 executa um laço que analisa todas as formigas e atualiza a melhor rota do ciclo. O passo 12 atualiza a melhor rota. Entre os passos 13 e 18 atualiza-se o feromônio. O passo 19 atualiza a visibilidade e o passo 20 esvazia a lista L^k que representa a rota da formiga k.

O algoritmo ACO-VNDP incorpora uma etapa de aperfeiçoamento da solução através do uso da técnica *Variable Neighborhood Descending with Perturbation* [Goldbarg et al. 2016]. A técnica citada é uma variante proposta por Mladenović e Hansen [1997] que examina a aplicação de diversas vizinhanças em um processo de busca, mesclando a busca com movimentos de perturbações denominados também de *shaking*. O procedimento VNDP consta no quadro 3.

O procedimento VNDP emprega o seguinte conjunto de vizinhanças: Insere vértice, onde para cada vértice que não pertence à rota é verificada a sua inserção no final da rota (vizinhança 1); Substituição, verifica-se todas as possíveis substituições de dois vértices da rota por vértices não pertencentes à rota (vizinhança 2); e 2-opt, onde todas as possibilidades de remover e reinserir duas arestas são verificadas (vizinhança 3). Em cada vizinhança considerada, preserva-se a melhor configuração alcançada. O procedimento *shaking* é representado pela troca de *z* posições vizinhas quaisquer da rota que são escolhidas aleatoriamente.

```
ACO ()
1.  NC = 0;
2.  Para cada aresta (i, j) τij = 0,1; Δτij = 0
3.  Enquanto (NC < NCmax)
4.      Definir as cidades iniciais das #m formigas
5.      Repita até que a quota mínima não seja atingida
6.        Para k = 1 até #m faça
7.            Mova a k-ésima formiga para localidade j
8.            Coloque a cidade j em L^k
9.      Para k = 1 até #m faça
10.         Calcule o custo de L^k
11.         Atualize a melhor_rota_ciclo
12.     Atualiza melhor_rota
13.     Para cada aresta (i, j)
14.         Para k = 1 até #m faça
15.             Se (i, j) ∈ Lk faça Δτij^k = Q/Lk
16.             Senão Δτij^k = 0
17.             Δτij = Δτij + Δτij^k
18.     Para cada aresta (i, j) τij = (1-ρ) τij + Δτij
19.     Para cada aresta (i, j) Δτij = 0
20.     NC ← NC + 1; esvaziar as listas L^k
```

Quadro 2 – Algoritmo ACO

```
VNDP (S, s')
1.  Selecione vizinhança Nv
2.  v ←1
3.  Enquanto v ≤ 3
4.      s' ← busca_local(Nv(S))
5.      Se f(s') < f(S)
6.          S ← s'
7.          v ←1
8.      Senão
9.          shaking(Nv(S), z)
10.         v ← v + 1
```

Quadro 3 – Procedimento VNDP

O algoritmo ACO-VNDP, apresentado no quadro 4, realiza inicialmente a definição do feromônio τ_{ij} e visibilidade $\Delta\tau_{ij}$ de todas as arestas. Logo após, dá sequência a um laço que é executado NC_{max} vezes, sendo que a cada ciclo define-se primeiramente a posição inicial das formigas e logo após, utilizando a probabilidade da equação (23). No passo 4, constroem-se #*m* formigas. Posteriormente, atualiza-se a melhor rota do ciclo e executa-se o procedimento VNDP, com o intuito de atualizar a melhor rota. Ao fim do ciclo, atualiza-se o feromônio, a visibilidade e esvazia-se a lista de solução L^k, que representa a rota realizada pela formiga k. O algoritmo ACO desenvolvido é similar ao ACO-VNDP com exceção do passo que se segue à linha 12.

```
ACO-VNDP ()
1.  NC = 0
2.  Para cada aresta (i, j) τij = 0,1; Δτij = 0
3.  Enquanto (NC < NCmax)
4.      Definir as cidades iniciais das #m formigas
5.      Repita até que a quota mínima não seja atingida
6.          Para k = 1 até #m faça
7.              Escolha a cidade j para mover a k-ésima formiga com (23)
8.              Coloque a cidade j em L^k
9.       Para k = 1 até #m faça
10.          Calcule o custo L^k
11.          Atualize a melhor_rota_ciclo
12.     VNDP (melhor_rota_ciclo, melhor_rota)
13.     Para cada aresta (i, j)
14.         Para k = 1 até #m faça
15.             Se (i, j) ∈ Lk façaΔτij^k = Q/Lk
16.             Senão Δτij^k = 0
17.             Δτij = Δτij + Δτij^k;
18.     Para cada aresta (i, j) τij = (1-ρ) τij + Δτij
19.     Para cada aresta (i, j) Δτij = 0
20.     NC ← NC + 1; esvaziar as listas L^k
```

Quadro 4 – Algoritmo ACO-VNDP

8. Resultados computacionais

Os algoritmos meta-heurísticos foram parametrizados com auxílio da ferramenta *IRACE - Iterated Race for Automatic Algorithm Configuration*, que realiza configuração automática de algoritmos [López-Ibánez et al. 2011]. Com base nas 24 instâncias criadas aleatoriamente para a parametrização, a ferramenta calculou, dentre as opções selecionadas pelos autores, os valores para os parâmetros sumarizados na tabela 2 e adotados no experimento.

Tabela 2 – Configurações dos algoritmos ACO (A) e ACO-VNDP (VND)

Parâmetro	Valores	A	VND	Parâmetro	Valores	A	VND
α	1,0 até 2,0	1,48	1,56	NC_{max} (VND)	30, 35, 40, 45, 50	-	50
β	3,0 até 4,0	3,85	3,85	*#m*	10, 15, 20, 25, 30	10	15
γ	5,0 até 6,0	5,85	5,99	ρ	0,1 até 1,0	0,96	0,15
NC_{max} (A)	100, 125, 150, 175, 200	125	-	*z*	2 até 4	-	2

O experimento com as heurísticas ACO e ACO-VNDP consistiu em 30 execuções de cada instância do conjunto de casos teste na mesma plataforma já relatada para os testes da seção 5. Os resultados são comparados qualitativamente através das melhores soluções obtidas com o algoritmo exato e da heurística *Ah*.

Os resultados para instâncias simétricas e assimétricas são apresentados, respectivamente, nas tabelas 3 e 4 que mostram o nome da instância (Instância), o valor da melhor solução encontrada (M), o valor mínimo encontrado pelo algoritmo da coluna (MIN), o valor médio (MED), o desvio padrão (D) e o tempo médio em segundos (TM).

Como pode-se observar na coluna MIN, os resultados qualitativos são conclusivamente favoráveis ao algoritmo ACO-VNDP. O tempo médio de execução é favorável ao ACO nas instâncias simétricas e assimétricas. Comportamento diferente ao verificado no tempo médio é constatado em relação ao valor médio e ao desvio padrão. O pequeno afastamento entre as soluções ótimas e as mínimas obtidas no ACO-VNDP é indicador de eficácia. Seus resultados superiores em relação às melhores soluções encontradas, mostradas na coluna M, confirmam seu potencial

também frente à heurística *ad hoc* proposta. Um experimento estatístico confirma as conclusões anteriormente citadas.

Tabela 3 – Resultados para as instâncias simétricas

		ACO				ACO-VNDP			
Instância	**M**	**MIN**	**MED**	**D**	**TM**	**MIN**	**MED**	**D**	**TM**
10-18-4-1	207,55	271,83	364,52	53,48	1,58	**207,55**	245,55	61,05	1,88
10-34-5-1	145,32	197,67	219,35	18,05	1,84	**154,40**	160,55	6,89	2,77
20-26-3-1	340,83	327,50	412,82	64,96	3,52	**275,50**	292,35	10,06	12,68
20-114-5-1*	238,58	204,83	254,20	40,84	5,06	**198,22**	204,61	1,21	9,47
30-101-3-1*	492,17	519,67	609,80	85,80	9,74	**411,08**	443,75	20,61	71,13
30-122-4-1*	392,45	379,07	457,52	53,17	10,26	**305,77**	342,25	19,46	62,45
10-14-3-2	338,33	472,67	764,96	178,09	1,52	**347,50**	365,90	11,74	2,60
10-18-4-2	344,00	467,33	843,41	108,79	1,59	**349,75**	363,59	14,63	2,06
20-26-3-2	677,42	1015,83	1268,39	195,77	3,50	**473,33**	510,53	28,15	18,66
20-114-5-2*	656,00	347,92	590,82	111,70	4,92	**284,17**	305,93	14,52	27,46
30-122-4-2*	1394,67	456,28	896,49	237,01	9,97	**428,70**	491,19	24,52	95,09
30-155-5-2*	942,67	408,63	542,59	83,40	11,95	**352,43**	383,75	15,94	71,29
10-14-3-3	292,75	459,33	513,06	80,74	1,58	**292,75**	298,93	12,89	2,34
10-34-5-3	154,83	259,75	274,01	32,72	1,76	**158,03**	177,87	7,70	3,35
20-47-4-3	554,28	525,95	665,32	65,50	3,93	**463,52**	500,99	19,89	12,62
20-114-5-3*	390,33	358,20	487,47	85,23	5,03	**239,83**	267,66	12,92	21,84
30-35-3-3	2087,00	884,50	1570,00	337,54	7,04	**766,00**	831,14	30,03	56,23
30-155-5-3*	930,33	442,00	597,09	81,42	10,35	**313,43**	374,31	22,44	90,90

Tabela 4 – Resultados para as instâncias assimétricas

		ACO				ACO-VNDP			
Instância	**M**	**MIN**	**MED**	**D**	**TM**	**MIN**	**MED**	**D**	**TM**
10-21-4-1	199,50	241,83	280,93	48,42	1,62	**205,20**	210,30	10,43	2,39
10-49-5-1*	150,33	226,08	321,41	49,17	2,11	**173,75**	194,03	19,04	3,43
20-26-3-1	361,58	403,92	577,50	153,24	3,65	**331,42**	370,17	29,73	12,16
20-95-5-1*	206,27	229,15	343,03	74,05	5,06	**193,25**	205,21	9,37	15,71
30-89-3-1*	405,67	407,83	548,32	73,85	9,05	**358,25**	367,57	9,42	22,40
30-56-4-1	881,50	578,47	894,98	169,51	8,15	**431,75**	472,36	19,76	71,38
10-17-3-2	329,50	784,00	848,10	47,64	1,55	**348,75**	362,56	13,18	2,93
10-21-4-2	398,45	426,75	539,96	63,79	1,62	**398,45**	401,34	9,41	2,35
20-26-3-2	431,00	505,42	740,02	205,18	3,61	**429,58**	462,26	26,08	15,65
20-95-5-2*	498,28	342,50	545,83	130,81	5,18	**294,27**	330,93	15,94	22,38
30-105-4-2*	512,90	495,08	630,93	77,08	10,23	**441,08**	485,38	24,19	66,39
30-143-5-2*	1010,17	547,82	704,26	103,75	11,14	**438,58**	492,11	23,50	69,65
10-17-3-3	294,42	406,25	539,39	91,46	1,56	**317,58**	321,13	3,08	2,64
10-24-5-3	189,30	490,00	490,00	0,00	1,70	**200,62**	200,62	0,00	3,06
20-42-4-3	454,70	416,92	588,26	102,09	3,79	**331,25**	360,53	16,21	16,67
20-38-5-3	470,98	674,40	959,61	167,98	3,56	**429,23**	459,29	16,02	21,41
30-50-3-3	1738,00	1041,50	1423,26	237,60	7,69	**720,17**	789,73	37,24	74,91
30-143-5-3*	930,33	442,00	597,09	81,42	10,35	**313,43**	374,31	22,44	90,90

9. Conclusão

Este trabalho apresentou o Problema do Caixeiro Viajante com Múltiplos Passageiros e Quota (PCV-MPQ). Foi apresentada uma formulação matemática para o problema a qual foi implementada em um *solver* que forneceu os primeiros resultados ótimos para 9 instâncias de um conjunto de 36 propostas neste trabalho. Foram apresentadas 3 heurísticas para o problema: uma

heurística *ad hoc*, uma baseada na meta-heurística Colônia de Formigas, e a terceira uma hibridização de Colônia de Formigas e Busca em Vizinhança Variável. A função de atualização da matriz de probabilidades dos algoritmos de formigas foi desenvolvida para este trabalho devido às particularidades do problema que considera uma rota, coleta de bônus e carregamento de passageiros em um veículo capacitado. O algoritmo híbrido apresentou os melhores resultados para o problema no que se refere à qualidade das soluções encontradas. Em trabalhos futuros serão aplicadas novas meta-heurísticas para o problema e serão investigadas hibridizações com métodos exatos.

Agradecimentos

Este trabalho foi financiado parcialmente pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), projetos 302387/2016-1 e 306702/2017-7.

Referências

Amey, A., Attanucci, J. & Mishalani, R. (2011). Real-time ridesharing–the opportunities and challenges of utilizing mobile phone technology to improve rideshare services. In *TRB Annual Meeting*, 1–17.

Ausiello, G., Demange, M., Laura, L. & Paschos, V. (2004). Algorithms for the on-line quota traveling salesman problem. In *International Computing and Combinatorics Conference*, 290–299.

Awerbuch, B., Azar, Y., Blum, A. & Vempala, S. (1998). New approximation guarantees for minimum-weight k-trees and prize-collecting salesman. *SIAM Journal on Computing* 28(1), 254–262.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks* 19(6), 621–636.

Berbeglia, G., Cordeau, J.-F. & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research* 202(1), 8–15.

Calheiros, Z. S. A. (2017). *O Problema do Caixeiro Viajante Com Passageiros*. Dissertação de mestrado, Pós-graduação em Sistemas e Computação, Universidade Federal do Rio Grande do Norte.

Chan, N. D. & Shaheen, S. A. (2012). Ridesharing in north america: Past, present, and future. *Transport Reviews* 32(1), 93–112.

Dailey, D., Loseff, D. & Meyers, D. (1999). Seattle smart traveler: dynamic ridematching on the world wide web. *Transportation Research Part C: Emerging Technologies* 7(1),17–32.

Deakin, E., Frick, K. & Shively, K. (2010). Markets for dynamic ridesharing? case of berkeley, california. *Transportation Research Record: Journal of the Transportation Research Board* 2187, 131–137.

Doerner, K. F. & Salazar-González, J.-J. (2014). Chapter 7: Pickup-and-delivery problems for people transportation. In *Vehicle Routing: Problems, Methods, and Applications*, Second Edition, 193–212. SIAM.

Dorigo, M. & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *IEEE CEC99 Congress on Evolutionary Computation* 2, 1470–1477.

Goldbarg, M. C. & Goldbarg, E. (2012). *Grafos: Conceitos, Algoritmos e Aplicações*. Elsevier.

Goldbarg, M. C., Goldbarg, E. F. G. & Luna, H. P. L. (2016). *Otimização Combinatória e Meta-heurísticas: Algoritmos e Aplicações*. Elsevier.

Gruebele, P. (2008). Interactive system for real time dynamic multi-hop carpooling. *Global Transport Knowledge Partnership*, 1-17. Disponível em http://www.dynamicridesharing.org/resources/Multi_hop_social_carpool_routing_System.pdf, último acesso em 4 de junho de 2018.

Kleiner, A., Nebel, B. & Ziparo, V. A. (2011). A mechanism for dynamic ride sharing based on parallel auctions. In *International Joint Conference on Artificial Intelligence* 11, 266–272.

Lin, S. & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21(2), 498–516.

López-Ibánez, M., Dubois-Lacoste, J., Stützle, T. & Birattari, M. (2011). *The IRACE Package, Iterated Race for Automatic Algorithm Configuration*. Technical Report TR/IRIDIA/2011- 004, IRIDIA, Université Libre de Bruxelles, Belgium.

Miller, C. E., Tucker, A. W. & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM* 7(4), 326–329.

Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research* 24(11), 1097–1100.

Morency, C. (2007). The ambivalence of ridesharing. *Transportation* 34(2), 239–253.

Nourinejad, M. (2014). *Dynamic Optimization Models for Ridesharing and Carsharing*. PhD thesis, MSc. Thesis, Department of Civil Engineering, University of Toronto.

Parragh, S. N., Doerner, K. F. & Hartl, R. F. (2008). A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft* 58(2), 81–117.

Yu, W., Liu, Z. & Bao, X. (2014). Optimal deterministic algorithms for some variants of online quota traveling salesman problem. *European Journal of Operational Research* 238(3), 735–740.