

Prize Collecting Traveling Salesman Problem with Ridesharing

Caixeiro Viajante com Coleta de Prêmios e Passageiros

Ygor Alcântara de Medeiros¹, Marco Cesar Goldberg^{1*}, Elizabeth Ferreira Gouvêa Goldberg¹

Abstract: The Prize Collecting Traveling Salesman Problem with Ridesharing is a model that joins elements from the Prize Collecting Traveling Salesman and the collaborative transport. The salesman is the driver of a capacitated vehicle and uses a ridesharing system to minimize travel costs. There are a penalty and a bonus associated with each vertex of a graph, G , that represents the problem. There is also a cost associated with each edge of G . The salesman must choose a subset of vertices to be visited so that the total bonus collection is at least a given parameter. The length of the tour plus the sum of penalties of all vertices not visited is as small as possible. There is a set of persons demanding rides. The ride request consists of a pickup and a drop off location, a maximum travel duration, and the maximum amount the person agrees to pay. The driver shares the cost associated with each arc in the tour with the passengers in the vehicle. Constraints from ride requests, as well as the capacity of the car, must be satisfied. We present a mathematical formulation for the problem investigated in this study and solve it in an optimization tool. We also present three heuristics that hybridize exact and heuristic methods. These algorithms use a decomposition strategy that other enriched vehicle routing problems can utilize.

Keywords: Travelling salesman Problem — Ridesharing — Metaheuristic

Resumo: O problema do Caixeiro Viajante com Coleta de Prêmios e Passageiros é um modelo que une elementos do clássico problema do Caixeiro Viajante com Coleta de Prêmios e do transporte colaborativo. O caixeiro é o motorista de um veículo capacitado e usa um sistema de ridesharing para minimizar seus custos de viagem. Existem uma penalidade e um bônus associados a cada vértice do grafo, G , que representa o problema. Também existe um custo associado a cada arco de G . O caixeiro deve escolher um subconjunto de vértices para serem visitados de forma que o total de bônus coletado nos vértices seja, no mínimo, um parâmetro especificado. O comprimento da rota mais a soma das penalidades associadas aos vértices não visitados deve ser o menor possível. Existe um conjunto de pessoas que demandam transporte. A requisição de transporte consiste dos pontos de embarque e desembarque, a duração máxima da viagem e o máximo que o passageiro concorda em pagar. O motorista divide o custo associado a cada arco com os passageiros no veículo. As restrições das requisições e da capacidade do carro devem ser satisfeitas. Apresentamos uma formulação matemática para o problema que é resolvida por uma ferramenta de otimização. Também, apresentamos três heurísticas que hibridizam métodos exatos com heurísticos. Estes algoritmos usam uma estratégia de decomposição que pode ser utilizada por outros problemas de roteamento enriquecido.

Palavras-Chave: Problema do Caixeiro Viajante — Ridesharing — Meta-heurística

¹ Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Natal, Rio Grande do Norte, Brasil

*Corresponding author: marcocgold@gmail.com

DOI: <http://dx.doi.org/10.22456/2175-2745.94082> • Received: 28/06/2019 • Accepted: 09/02/2020

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introdução

O objetivo deste trabalho é introduzir o Problema do Caixeiro Viajante com Coleta de Prêmios e Passageiros (PCV-CPP), ou *Prize Collecting Traveling Salesman Problem with Ridesharing* (PCTSP-R), apresentando uma formulação matemática e algoritmos heurísticos.

O modelo matemático proposto neste trabalho para o PCV-CPP tem por base a formulação de Balas [1] para o Problema do Caixeiro Viajante com Coleta de Prêmios (PCVCP). O modelo do PCV-CPP expande a formulação citada incorporando opções do tipo *ridesharing* de forma a mesclar o transporte colaborativo com a exigência de coleta de prêmios visando reduzir os custos de rota do caixeiro. O embarque de

passageiros é seletivo, não se impondo qualquer atendimento obrigatório de demanda. São garantidas as condições de tarifa máxima fixadas pelos passageiros. O modelo pode potencializar a sinergia entre os tradicionais sistemas de roteamento de veículos com tarefas e o transporte solidário.

A importância do transporte solidário, abrigado na literatura sob a nomeação de *ridesharing*, se dá em várias dimensões. O transporte solidário pode diminuir a poluição no meio ambiente e reduzir fluxos de carros nas vias de transporte, com a consequente redução de engarrafamentos nas metrópoles e melhoria da qualidade de vida. Adicionalmente, são visíveis os ganhos no conforto do passageiro, na socialização, na redução dos custos operacionais do sistema, na redução das áreas de estacionamento de veículos e até nos investimentos públicos para implantação e manutenção de sistemas de transportes [15].

No PCV-CPP, o caixeiro pode embarcar, dentre os vários passageiros de oportunidade disponíveis em cada localidade potencialmente visitável, aqueles cuja origem e destino estejam no conjunto das localidades selecionadas pelo caixeiro. Em função do desembarque, o destino dos passageiros embarcados deve pertencer ao trecho da rota ainda por visitar pelo caixeiro. O embarque só é possível se a parcela do rateio atribuída ao passageiro em função de seu deslocamento, for menor ou igual a um valor máximo previamente estabelecido pelo próprio passageiro. O cálculo do rateio depende do conhecimento da rota e de todos os embarques e desembarques da rota. Por fim, a capacidade do carro não pode ser ultrapassada.

Os custos de viagem são rateados, trecho a trecho, entre todos os ocupantes do carro, computando-se o caixeiro como um dos ocupantes. Portanto, o modelo proíbe que o caixeiro lucre com o compartilhamento de assentos, descaracterizando-se como atividade lucrativa. Como presentemente formulado, os passageiros participam de um transporte solidário, não de transporte tarifado. O caixeiro pode reduzir seus custos através do transporte solidário. Entretanto, continua obrigado por força de seu trabalho, a coletar um valor mínimo de prêmios nas localidades de sua rota e sofrendo uma penalidade específica em função de cada demanda que deixa de atender.

O PCV-CPP é um modelo de embarque e desembarque seletivo, cuja demanda de passageiros é atendida de forma oportunista, exclusivamente em função do interesse do caixeiro. A função objetivo da formulação contabiliza somente os custos do caixeiro. A escolha das tarefas que serão realizadas pelo caixeiro, modeladas através da coleta dos prêmios nas localidades é, igualmente, uma decisão seletiva.

O modelo objeto da presente pesquisa, por envolver várias decisões, além de ser uma variante do problema de roteamento de veículos com coleta de prêmios [1], encontra interfaces com vários modelos da literatura que otimizam o embarque e desembarque seletivo de passageiros [8] e com modelos de *ridesharing* [15]. Por objetividade e pela diversidade de modelos que compartilham elementos em comum com PCV-CPP, serão abordados os mais recentes trabalhos diretamente

relacionados.

Calheiros [3] formulou o Problema do Caixeiro Viajante (PCV) com Passageiros sem considerar a coleta de bônus. O trabalho desenvolve algoritmos e experimento computacional para a solução do caso particular de apenas um passageiro demandar embarque em cada localidade. Bastos et al. [2] desenvolveram uma formulação e algoritmos para uma variante do modelo abordado por Calheiros [3] considerando, adicionalmente, a existência de pedágios em algumas das ligações entre as localidades. Bastos et al. [2] trataram os pedágios sob o conceito de faixas de tráfego associadas à descontos ou isenções de pedágios (*high occupancy vehicle lanes*), bem como facilidades de circulação decorrentes do carro atender a uma taxa de ocupação mínima previamente especificada. Sua experimentação computacional igualmente considera a demanda de um passageiro por localidade. Sabry et al. [16] desenvolveram uma formulação e algoritmos para uma variante do modelo Caixeiro Viajante Alugador com Passageiros. O trabalho de Sabry et al. [16] mesclou o problema do caixeiro viajante com passageiros formulado pelo modelo de Calheiros [3] com o problema do caixeiro alugador, segundo a formulação de Goldbarg et al. [7], mantendo a experimentação algorítmica com somente um passageiro por localidade. Finalmente, Silva [17] desenvolveu uma formulação e algoritmos para o Caixeiro Viajante com Passageiros e Quota (PCV-PQ). O modelo do PCV-PQ de Silva [17] distingue-se do modelo de Calheiros [3], todavia, como nos demais trabalhos anteriormente citados, mantém o estudo particular do caso de um passageiro por localidade. Adicionalmente, por se tratar de um problema de cota, a função objetivo do modelo não é penalizada quando a rota deixa de visitar localidades que demandam coleta.

O presente trabalho apresenta, além da formulação do PCV-CPP, algoritmos heurísticos para o problema. Na etapa de validação dos algoritmos, desenvolve-se um estudo estatístico para a solução de um banco de instâncias proposto para o problema. Para ancorar os resultados alcançados pelos algoritmos que utilizam passos heurísticos em alguma de suas etapas, são desenvolvidas duas heurísticas que desempenham exclusivamente passos exatos de solução. No caso, solucionando de forma ótima o passo da construção da rota e o passo da realização do embarque de passageiros. As instâncias de menor porte são solucionadas de forma exata por um *software* de otimização. A seção 2 apresenta a formulação matemática para o PCV-CPP. As seções 3 e 4 descrevem os algoritmos desenvolvidos. A seção 5 apresenta os resultados do experimento computacional, incluindo a metodologia utilizada na criação das instâncias. A seção 6 apresenta as conclusões.

2. Formulação Matemática

Seja o grafo $G = (N, M, B, P)$, onde $N = \{1, \dots, n\}$ é o conjunto de vértices ou localidades, M representa o conjunto de arestas que interligam os vértices, $B = \{b_1, \dots, b_n\}$ corresponde ao conjunto dos prêmios que são associados aos

vértices $i \in N$, e $P = \{p_1, \dots, p_n\}$ é o conjunto de penalidades associadas aos vértices. Quando a rota do caixeiro visita uma localidade, o bônus associado a essa localidade é considerado coletado. É considerado que o caixeiro parte da cidade 1 e retorna a ela. Como a localidade 1 é visita obrigatória, o valor de seu suposto bônus é igual a zero, $b_1 = 0$. Caso a localidade i deixe de ser visitada pelo caixeiro, uma penalidade p_i é somada à parcela de custo na função objetivo.

São considerados r passageiros demandando transporte e distribuídos por $|N| = n$ localidades potencialmente visitáveis pelo caixeiro, $r \geq n$. O número de passageiros demandando por acentos no carro pode variar em cada localidade, sendo maior ou igual a zero.

O conjunto $L = \{l_1, \dots, l_r\}$ define os passageiros. Cada passageiro l_i admite gastar, no máximo, t_i unidades monetárias para compartilhar as despesas de viagem. A despesa de cada passageiro é calculada pelo rateio entre os ocupantes do veículo, somado trecho a trecho de sua viagem, desde o embarque até o desembarque. O caixeiro viajante é computado no rateio como se passageiro fosse. s_i são as localidades de embarque dos passageiros l_i , e d_i são as localidades de seu desembarque, considerando-se $l_i \neq d_i$, $i = 1, \dots, r$. Denomina-se por s o índice da localidade de início e término da rota. O parâmetro B_{min} quantifica o valor de prêmio mínimo que o caixeiro viajante necessita recolher durante o seu trajeto. O coeficiente c_{ij} corresponde ao custo da aresta $(i, j) \in M$.

Considerando as notações anteriores, as variáveis de decisão para este problema estão definidas a seguir.

- x_{ij} : variável binária que define se a aresta $(i, j) \in M$ pertence a rota do caixeiro viajante ($x_{ij} = 1$) ou não ($x_{ij} = 0$).
- v_{ij}^l : variável binária que define se a aresta $(i, j) \in M$ pertence a rota do passageiro de índice l , $l = 1, \dots, r$, ($v_{ij}^l = 1$) ou não ($v_{ij}^l = 0$).
- y_i : variável binária que define se o vértice $i \in N$ pertence a rota do caixeiro viajante ($y_i = 1$) ou não ($y_i = 0$).
- u_i : variável inteira que representa a ordem do vértice i na rota do caixeiro viajante.

O modelo matemático deste trabalho adapta o proposto por Silva [17] para as condições do PCV-CPP de forma a considerar as penalidades $P = p_1, \dots, p_n$ decorrentes da ausência de uma visita em qualquer localidade $i \in N$.

$$\min Z = \sum_{(i,j) \in M} \frac{c_{ij}x_{ij}}{1 + \sum_{l \in L} v_{ij}^l} + \sum_{i \in N} p_i(1 - y_i) \quad (1)$$

Sujeito a:

$$\sum_{j \in N \setminus \{i\}} x_{ij} = y_i, \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N \setminus \{i\}} x_{ji} = y_i, \quad \forall i \in N \quad (3)$$

$$\sum_{i \in N \setminus \{j\}} x_{ij} \leq 1, \quad \forall i \in N \setminus \{s\} \quad (4)$$

$$\sum_{i \in N \setminus \{j\}} x_{ji} \leq 1, \quad \forall i \in N \setminus \{s\} \quad (5)$$

$$\sum_{i \in N \setminus \{s\}} x_{si} = 1 \quad (6)$$

$$\sum_{i \in N \setminus \{s\}} x_{is} = 1 \quad (7)$$

$$\sum_{j \in N \setminus \{i\}} x_{ij} = \sum_{j \in N \setminus \{i\}} x_{ji}, \quad \forall j \in N \setminus \{s\} \quad (8)$$

$$u_i - u_j + (n-1)x_{ij} \leq n-2, \quad 2 \leq i, j \leq n, i \neq j \quad (9)$$

$$\sum_{i \in N} b_i y_i \geq b_{min} \quad (10)$$

$$\sum_{i \in L} v_{ij}^l \leq kx_{ij}, \quad \forall (i, j) \in M, i \neq j \quad (11)$$

$$\sum_{(i,j) \in M} \frac{c_{ij}v_{ij}^l}{1 + \sum_{l \in L} v_{ij}^l} - t_l \leq 0, \quad l = 1, \dots, r, i \neq j \quad (12)$$

$$\sum_{j \in N \setminus \{i\}} v_{ji}^l - \sum_{j \in N \setminus \{i\}} v_{ij}^l = 0, \quad l = 1, \dots, r, i \in N \setminus \{s_l, d_l\} \quad (13)$$

$$\sum_{i \in N \setminus \{s_l\}} v_{is_l}^l = 0, \quad l = 1, \dots, r \quad (14)$$

$$\sum_{i \in N \setminus \{d_l\}} v_{d_l i}^l = 0, \quad l = 1, \dots, r \quad (15)$$

$$\sum_{i \in N \setminus \{s\}} v_{s_i}^l = 0, \quad l = 1, \dots, r, s \neq s_l \quad (16)$$

$$1 \leq u_i \leq n-1 \quad 2 \leq i \leq n \quad (17)$$

$$b_s = 0 \quad (18)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in M \quad (19)$$

$$v_{ij}^l \in \{0, 1\} \quad \forall (i, j) \in M, l = 1, \dots, r \quad (20)$$

$$u_i \in \mathbb{N}, \quad \forall i \in N \quad (21)$$

$$y_i \in \{0, 1\}, \quad \forall i \in N \quad (22)$$

A função objetivo apresentada pela equação (1), contabiliza o custo total do caixeiro viajante, considerando a soma da parcela do rateio de rota com a parcela das penalidades

decorrente da eventual ausência de visitas em localidades do grafo. As restrições (2) e (3) controlam os valores da variável y_i que recebe o valor 1 quando a localidade i pertence à rota. As restrições (4) e (5) permitem que o percurso do caixeiro possa ser formado por um subconjunto dos vértices pertencentes a N . As restrições (6) e (7) asseguram que o trajeto do caixeiro inicia e termina no vértice inicial $s = 1$. A restrição (8) garante que a rota do caixeiro possui continuidade. A restrição (9) garante a formação de um único ciclo na solução, conforme a proposta de Miller et al. [13]. A restrição (10) garante que o caixeiro vai coletar a quantidade mínima de prêmios B_{min} imposta como tarefa. A restrição (11) garante que a capacidade do carro não será excedida. A restrição (12) assegura que o passageiro de índice l , $l = 1, \dots, r$, embarcado no carro, só dispendirá no rateio o valor máximo t_l unidades monetárias. A restrição (13) obriga que o passageiro l só desembarque no seu vértice de destino d_l . As restrições (14) e (15) asseguram que os passageiros de índice l , $l = 1, \dots, r$, não poderão desembarcar no seu vértice de embarque s_l e não poderão embarcar no seu vértice de desembarque d_l . A restrição (16) restringe que os passageiros desembarcados na localidade inicial da rota, embarquem novamente. A restrição (17) define o intervalo de valor da variável auxiliar u para assegurar o funcionamento da restrição (9). A restrição (18) determina que o vértice inicial s não possui prêmio de coleta. As restrições (19) a (22) caracterizam o intervalo de valor das variáveis de decisão do modelo.

3. Algoritmos HERS na solução do PCV-CPP

Os algoritmos propostos são denominados Algoritmos com Etapas Heurísticas, Exatas e de Roleta (HERS – *Heuristic, Exact and Roulette Steps*). Os citados algoritmos propõem executar a divisão da construção e aperfeiçoamento da solução de um problema de otimização em diversas fases. No caso da presente pesquisa, as heurísticas propostas dividem o processo construtivo em três diferentes etapas, mesclando os objetivos de intensificação e diversificação entre essas fases. Na abordagem proposta, não há qualquer divisão de fases que separe intenções de intensificação ou de diversificação da busca como, por exemplo, na meta-heurística construtiva GRASP [5]. Igualmente, são definidas livremente tanto o número de fases quanto as etapas com buscas locais. Como um exemplo possível para a proposta geral de tipo de algoritmos, a primeira fase do processo construtivo pode ser desempenhada por um procedimento exato, seguindo-se por uma segunda fase de construção realizada através de uma roleta [11] e, finalmente, concluindo-se a construção da solução através de um segundo procedimento exato.

Em princípio, cada diferente estratégia construtiva estará associada a um diferente tipo de variável de decisão. A decomposição da solução construtiva em fases e a aplicação de diferentes estratégias construtivas em cada fase caracterizam os algoritmos HERS. Tais algoritmos podem ser aplicados a diferentes problemas de otimização combinatória com

decisões acopladas. O algoritmo 1 exhibe a decomposição da tomada de decisão do método, ressaltando-se que cada fase implementa uma estratégia específica para definir suas variáveis associadas, todavia embutindo as decisões alcançadas no processo da próxima fase. As variáveis Sol_{aA} , Sol_{bAB} e $Sol_{pAB \dots P}$ denotam, respectivamente, as soluções obtidas pelas estratégias construtivas A , B , e P .

Algoritmo 1: HERS

Entrada: decisões $A, B, \dots P$

Saída: Sol

```

1 início
2    $Sol_{aA} \leftarrow \text{Solver}(\text{Model}, A)$ 
3    $Sol_{bAB} \leftarrow \text{Solver}(\text{Model}, Sol_{aA})$ 
4    $Sol_{pAB \dots P} \leftarrow$ 
      $\text{Solver}(\text{Model}, Sol_{pAB \dots P} - 1)$ 
5    $Sol \leftarrow Sol_{pAB \dots P}$ 
6   retorna  $Sol$ 
7 fim
```

Como abordado no modelo da seção 2, a solução do PCV-CPP deve envolver, de forma ótima e simultânea, as seguintes decisões: 1- A seleção das localidades que serão visitadas pela rota do caixeiro; 2- A rota que será cumprida para visitar as localidades selecionadas; 3- A estratégia de embarque dos passageiros.

O bônus é considerado automaticamente coletado em cada localidade visitada. As três decisões do problema são interferentes, contudo, podem ser realizadas de modo independente. É possível decidir, primeiramente, quais localidades serão visitadas pelo caixeiro. Após, é possível determinar uma rota que passe pelas localidades. Finalmente, um embarque de passageiros pode ser definido para a rota formada nos dois primeiros passos. O processo construtivo deve, todavia, tomar as decisões de modo a garantir formar, ao final, uma solução viável. O embarque depende do conhecimento das localidades e da rota. A rota depende do conhecimento prévio das localidades visitadas. As localidades podem ser selecionadas exclusivamente em função do atendimento do recolhimento de bônus. Atendida a hierarquia de construção, no caso, é possível a formação de soluções viáveis. Cada uma das etapas construtivas, contudo, pode ser realizada através de diferentes estratégias.

HERS1 (algoritmo 2) e HERS2 são duas heurísticas HERS para o PCV-CPP. Ambas constroem uma solução em três etapas. A primeira seleciona um conjunto de localidades formando uma rota viável. A etapa é realizada com o uso de uma roleta [11] ponderada pela penalidade das localidades, sorteando-se localidades até obter um conjunto de localidades que atenda ao prêmio mínimo obrigatório. A primeira etapa é resumida nos passos de 6 até 8 do algoritmo 2.

A segunda etapa construtiva define a rota que o caixeiro vai cumprir sobre o conjunto das R localidades selecionadas. A decisão é alcançada no passo 9 através da aplicação da

Algoritmo 2: HERS1

Entrada: $G = (N, M, B, P)$, B_{min} , Passageiros L ,
 $Itermax$

Saída: Sol^*

```

1 início
2    $Iter \leftarrow 0$ ;
3   enquanto  $Iter < Itermax$  faça
4     /*  $R$  é um conjunto de
5       localidades */
6      $R \leftarrow R \cup \{0\}$ ;
7      $Candidatos \leftarrow N \setminus \{0\}$ ;
8     enquanto  $bônus$  de  $R < B_{min}$  faça
9        $c \leftarrow v \in Candidatos$  via roleta;
10       $Candidatos \leftarrow Candidatos \setminus \{c\}$ ;
11       $R \leftarrow R \cup \{c\}$ ;
12    fim_enquanto
13     $Sol \leftarrow LKH(R)$ ;
14     $Sol \leftarrow EmbarqueHeurístico(Sol, L)$ ;
15    /*  $BestSols$  contém  $SolMax$ 
16      melhores soluções */
17     $Bestsols \leftarrow MelhoresSol(Bestsols, Sol)$ ;
18     $Sol^* \leftarrow$  A melhor solução em  $Bestsols$ ;
19    se  $f(Sol) < f(Sol^*)$  então
20       $Iter \leftarrow 0$ ;
21    fim_se
22  fim_enquanto
23  para cada  $Sol' \in BestSols$  faça
24     $Sol' \leftarrow EmbarqueExato(Sol', L)$ ;
25     $BestSols \leftarrow MelhoresSol(Bestsols, Sol')$ ;
26  fim_para
27   $Sol^* \leftarrow$  A melhor solução em  $Bestsols$ ;
28  retorna  $Sol^*$ ;
29 fim

```

heurística Lin-Kernighan [10]. A terceira fase realiza um embarque heurístico dos passageiros no passo 10. O procedimento heurístico de embarque é apresentado no algoritmo 3.

A heurística HERS1, após o passo 16, desconsidera o embarque da melhor solução heurística encontrada e desenvolve uma solução exata para o embarque, concluindo a solução. A heurística HERS1 termina após a solução exata do embarque.

A heurística apresentada no algoritmo 3 leva em consideração a restrição de tarifa imposta pelo passageiro e a capacidade do veículo. A heurística cria uma fila de passageiros a serem embarcados, seguindo a ordem de visita das localidades na rota. Os passageiros são inseridos nesta fila se e somente se:

1. As localidades de origem e destino do passageiro pertencerem às visitadas na rota do motorista.
2. A localidade de destino situar-se, na rota, adiante da

Algoritmo 3: Procedimento EmbarqueHeurístico

Entrada: Solução S , Passageiros L

Saída: S

```

1 início
2   Inicialize uma fila  $f$  com todos os passageiros
3    $l \in L$ , se e somente se,  $l_{origem} \in S.rota$  e
4    $S.rota.indice(l_{destino}) > S.rota.indice(l_{origem})$ ;
5   enquanto  $f \neq \emptyset$  faça
6      $l \leftarrow f.pop()$ ;
7     se houver vaga no carro entre o trajeto do
8     passageiro  $l$  então
9       Marque  $l \in L$  como embarcado;
10    fim_se
11  fim_enquanto
12  para todo passageiro embarcado  $l \in L$  faça
13     $Custo \leftarrow$  Custo do trajeto do passageiro  $l$  na
14    rota da solução  $S$ ;
15    se  $Custo > l.limite$  então
16      Desmarque  $l \in L$  como embarcado;
17    fim_se
18  fim_para
19   $S \leftarrow S \cup L$ ;
20  retorna  $S$ ;
21 fim

```

localidade de embarque.

Garantido a viabilidade do desembarque, o algoritmo busca embarcar os passageiros seguindo a ordem estabelecida pela fila. A primeira verificação testa a capacidade do carro relacionado ao trajeto de cada passageiro. A segunda verificação garante a tarifa limite imposta pelo passageiro na medida em que o rateio se forma.

A heurística HERS2 diferencia-se da HERS1 por submeter a solução Sol obtida no passo 10 do algoritmo 2 a um procedimento de busca local segundo a arquitetura VNS (*Variable Neighborhood Search*). A heurística VNS foi introduzida por Mladenović e Hansen [14]. Para o PCV-CPP, são usadas três diferentes vizinhanças descritas a seguir. A sequência de aplicação das três vizinhanças é descrita no algoritmo 4.

- Vizinhança $N^1(Sol)$: Obtida pela troca de 2 vértices adjacentes na rota por 2 vértices que não pertencem ao trajeto.
- Vizinhança $N^2(Sol)$: Obtida pela remoção de 2 vértices adjacentes no trajeto por 1 vértice que não pertence ao trajeto, na mesma posição dos vértices removidos.
- Vizinhança $N^3(Sol)$: Obtida pela remoção de 1 vértice no trajeto do caixeiro e inserção de 3 vértices não pertencem ao trajeto, na mesma posição que o vértice removido.

Algoritmo 4: VNS para o PCV-CPP

Entrada: $BestSol$, Passageiros L
Saída: $BestSol$

```

1 início
2    $Sol \leftarrow BestSol$ ;
3    $Iter \leftarrow 0$ ;
4   enquanto  $Iter < Itermax$  faça
5      $r \leftarrow 3$ ;
6      $k \leftarrow 0$ ;
7     enquanto  $k < r$  faça
8       Gere um vizinho qualquer  $Sol' \in N^k(Sol)$ ;
9        $Sol'' \leftarrow LKH(Sol)$ ;
10      EmbarqueHeurístico( $Sol''$ ,  $L$ );
11      se  $f(Sol'') < f(Sol)$  então
12         $Sol \leftarrow Sol''$ ;
13         $k \leftarrow 1$ ;
14        senão  $\{k \leftarrow k + 1\}$ ;
15      fim_se
16    fim_enquanto
17    se  $f(Sol) < f(BestSol)$  então
18       $BestSol \leftarrow Sol$ ;
19       $f(BestSol) \leftarrow f(Sol)$ ;
20       $Iter \leftarrow 0$ ;
21      senão  $\{Iter \leftarrow Iter + 1\}$ ;
22    fim_se
23  fim_enquanto
24  retorna  $BestSol$ ;
25 fim
```

3.1 Heurísticas com Passos Exatos (HES - Heuristics of Exact Steps)

Com base na arquitetura proposta no algoritmo 1, é possível propor duas heurísticas que construam soluções para o problema desempenhando algoritmos exatos em cada etapa. As duas estão resumidas nos algoritmos 5 e 6.

Algoritmo 5: HES/CPP

Entrada: $G = (N, M, B, P)$, B_{min} , Passageiros L
Saída: Sol

```

1 início
2    $Sol.rota \leftarrow SolverRotaModel_1(G, B_{min})$ ;
3    $Sol \leftarrow SolverCarregamento(Sol.rota, L)$ ;
4   retorna  $Sol$ ;
5 fim
```

A heurística denominada CPP, no passo $SolverRota$, soluciona de forma exata Model_1, usando o modelo de programação matemática de coleta de prêmios proposto por Balas [1]. A heurística PCV soluciona de forma exata Model_2, que corresponde ao modelo de programação matemática do caixeiro viajante, conforme formulação proposta por Miller et al. [13].

Algoritmo 6: HES/PCV

Entrada: $G = (N, M, B, P)$, B_{min} , Passageiros L
Saída: Sol

```

1 início
2    $Sol.rota \leftarrow SolverRotaModel_2(G)$ ;
3    $Sol \leftarrow SolverCarregamento(Sol.rota, L)$ ;
4   retorna  $Sol$ ;
5 fim
```

Deve-se observar que a solução do caixeiro viajante obtida na heurística PCV é também uma solução para o problema de coleta de prêmios, uma vez que, ao visitar todas as localidades, atende obrigatoriamente à coleta da cota mínima B_{min} do modelo de Balas [1] para qualquer instância PCV-CPP viável. Por outro lado, a solução do caixeiro viajante resulta na neutralização do efeito da penalidade na função objetivo do modelo do caixeiro com coleta de prêmios. A redução da parcela de penalidades pode ser potencialmente promissora para gerar bons limites para o problema, especialmente em um modelo que possui a possibilidade de reduzir os custos nominais das ligações da rota através do rateio de despesas. A redução devida aos ganhos de embarque tem potencial para, em alguns casos, alterar a economia de rota decorrente da redução do número de localidades na rota. Usando o mesmo raciocínio, não é possível descartar o interesse de realizar visitas para adiante do valor mínimo imposto pela restrição de coleta de prêmios e assim explorar melhores configurações de embarque.

No caso das heurísticas HES/PCV que solucionam o caixeiro viajante, a fase de escolha das localidades se funde com a fase da escolha das rotas. Segue-se, depois da determinação da rota do caixeiro viajante, diretamente o embarque dos passageiros na rota encontrada.

A segunda fase das heurísticas descritas no algoritmo 5 realiza a solução exata do embarque de veículos na solução parcial construída na primeira fase. Um modelo de programação capaz de resolver essa etapa considera a variável de decisão $v_j \in [0, 1]$ que assume o valor 1 se o passageiro $j = 1, \dots, r$ está embarcado na solução e 0 em caso contrário, a constante $C_i \in \mathbb{R}^+$, que define o custo da i -ésima ligação na rota e , por fim, a constante $t_j \in \mathbb{R}^+$, $j = 1, \dots, r$, o valor da tarifa admissível para o passageiro j . O passo que soluciona de forma exata o carregamento das rotas, $SolverCarregamento$, pode ser formulado pelo modelo que está representado nas equações 23 a 28 e submetido para a solução exata em um solver.

$$\min Z = \sum_{i \in M} C_i - \sum_{w=2}^k \frac{C_i z_{iw}}{w(w-1)} \quad (23)$$

Sujeito a:

$$\sum_{\{j \in L | org(j) \leq i < dst(j)\}} v_j = \sum_{w=2}^k z_{iw}, \quad \forall i \in N \quad (24)$$

$$\sum_{i \in N} q_{li} \leq t_l, \quad \forall l \in L \quad (25)$$

$$c_i v_j - \sum_{w=2}^k \frac{c_i z_{iw}}{w(w-1)} \leq q_{ji}, \quad \forall i \in M, \{\forall j \in L, \text{org}(j) \leq i < \text{dst}(j)\} \quad (26)$$

$$q_{li} \in \mathbb{R}^+, \quad \forall i \in N, \forall l \in L \quad (27)$$

$$z_{iw} < 1, z_{iw} \in \mathbb{R}^+, \quad \forall i \in N, 2 \leq w \leq k \quad (28)$$

Observe-se que, ao realizar passos de solução exata, as heurísticas HES podem enfrentar a solução de problemas NP-difíceis. O fato sugere um esforço computacional, para um caso geral, de ordem exponencial no tamanho do número de variáveis da etapa. Assim, as heurísticas HES não garantem, para os casos reais, a certeza de seu uso generalizado. Presentemente, a estratégia mostrou-se justificada e viável, permitindo atender a necessidade de buscar boas soluções para o conjunto de instâncias de teste e estabelecer uma ancoragem qualitativa para as soluções obtidas pelos procedimentos HERS propostos. Adicionalmente, a composição de soluções exatas permite examinar o potencial de acoplamento das decisões na instância atacada, testando o afastamento do ótimo global do valor obtido por uma eventual composição de ótimos locais. Como o problema PCV-CPP é de extrema complexidade, contendo restrições e função objetivo não lineares, mesmo casos de pequeno porte são de difícil solução. Contudo, quando decomposto em etapas com decisões independentes, o porte do problema é suficiente para viabilizar, computacionalmente, a solução exata das etapas construtivas.

4. GRASP

A meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*), apresentada por Feo e Resende [5], realiza um processo dividido em duas fases. A primeira fase constrói uma solução e a segunda aperfeiçoa a solução construída.

O algoritmo 7 mostra os passos do algoritmo GRASP para a solução do PCV-CPP. Os passos de 6 a 11 implementam o método construtivo. A etapa de 6 a 9 implementa uma escolha pelo método da roleta [11], ponderada pela penalidade das localidades, sorteando-se as localidades para formar um conjunto de visitas que atenda ao prêmio mínimo necessário. Em seguida, é utilizada a heurística Lin-Kernighan [10], com a finalidade de obter a ordem de visita sobre o conjunto de localidades obtidas pelo passo anterior.

O aspecto adaptativo do GRASP é localizado na segunda etapa de construção da solução PCV-CPP. Como pode ser visualizado a seguir pelo algoritmo 8.

A heurística do algoritmo 8 é do tipo semigulosa aleatória, onde se leva em consideração a restrição de tarifa imposta

Algoritmo 7: GRASP para o PCV-CPP

Entrada: $G = (N, M, B, P)$, B_{min} , Passageiros L , $Itermax$

Saída: $BestSol$

/ R é um conjunto de localidades
/

```

1 início
2   Iter ← 0;
3   enquanto Iter < Itermax faça
4     R ← R ∪ {0};
5     Candidatos ← N \ {0};
6     enquanto bônus de R < Bmin faça
7       c ← v ∈ Candidatos via roleta;
8       Candidatos ← Candidatos \ {c};
9       R ← R ∪ {c};
10    fim_enquanto
11    Sol ← LKH(R);
12    Sol ← EmbarqueAdaptativo(Sol, L);
13    Sol ← VNS(Sol);
14    BestSols ← MelhoresSol(BestSols, Sol);
15    Sol* ← A melhor solução em BestSols;
16    se f(Sol) < f(Sol*) então
17      Iter ← 0;
18    fim_se
19  fim_enquanto
20  para cada Sol' ∈ BestSols faça
21    Sol' ← EmbarqueExato(Sol', L);
22    BestSols ← MelhoresSol(BestSols, Sol');
23  fim_para
24  Sol* ← A melhor solução em BestSols;
25  retorna Sol*;

```

pelo passageiro e a capacidade do veículo. O aspecto adaptativo se dá como base na utilização de uma Lista Restrita de Candidatos (LRC) formada pelos passageiros a serem embarcados.

Inicialmente é criado um conjunto de candidatos C com todos os passageiros aptos a embarcar no veículo do caixeiro. O passageiro é inserido no conjunto de candidatos C , se e somente se, a localidade de origem e de destino do passageiro considerado pertencer às localidades visitadas na rota do caixeiro e se, na rota, a localidade de destino for localizada adiante da localidade de embarque.

A função $g(c)$ define quais passageiros do conjunto de candidatos C pertencem à LRC. Desta forma, a função utiliza como valor, a diferença do ponto de origem e destino do passageiro na rota do caixeiro, ou seja, a quantidade de localidades que o passageiro visitará embarcado no veículo. Para esta função são utilizadas equações 29 a 31.

$$g' = \max\{g(c) | c \in C\} \quad (29)$$

Algoritmo 8: Heurística de Embarque Adaptativo**Entrada:** Solução S , Passageiros L **Saída:** S

```

1 início
2   Inicialize uma lista de candidatos  $C$  com todos os
   passageiros  $l \in L$ , se e somente se,  $l_{origem}$  e
    $l_{destino} \in S.rota$  e
    $S.rota.indice(l_{destino}) > S.rota.indice(l_{origem})$ ;
3   enquanto  $C \neq \emptyset$  faça
4     Para todo  $c \in C$  compute o valor da função
      $g(c)$  da forma que se segue,
      $S.rota.indice(c_{destino}) -$ 
      $S.rota.indice(c_{origem})$ ;
5     Defina  $LRC(C) \leftarrow \{c \in C | g(c)\}$ ;
6     Selecione  $c^*$  aleatoriamente de  $LRC(C)$ ;
7     se houver vaga no carro entre o trajeto do
     passageiro  $c^*$  então
8       Marque  $l \in L$  referente à  $c^*$  como
       embarcado;
9     fim_se
10    Remova  $c^*$  dos passageiros candidatos  $C$ ;
11  fim_enquanto
12  para todo passageiro embarcado  $l \in L$  faça
13     $Custo \leftarrow$  Custo do trajeto do passageiro  $l$  na
    rota da solução  $S$ ;
14    se  $Custo > l.limite$  então
15      Desmarque  $l \in L$  como embarcado;
16    fim_se
17  fim_para
18   $S \leftarrow S \cup L$ ;
19  retorna  $S$ ;
20 fim

```

$$g^* = \min\{g(c) | c \in C\} \quad (30)$$

$$g^* \leq g(c) \leq g^* + \alpha(g' - g^*) \quad (31)$$

No passo 5 do algoritmo 8 é definida a LRC com base na função $g(c)$. Desta forma, o passageiro candidato $c \in C$ é inserido na LRC, se e somente se, a equação 31 for válida. Vale salientar que a equação 31 possui o parâmetro α que é passado como parâmetro ao algoritmo. Em seguida, pelo passo 6, o algoritmo seleciona aleatoriamente um passageiro que pertença a LRC para embarcar no veículo. A primeira verificação é da capacidade do carro relacionado ao trajeto de cada passageiro. A segunda verificação visa garantir o limite de tarifa do passageiro considerando o rateio do custo que é formado em cada trecho da rota.

A etapa de intensificação de busca do GRASP é realizada pelo VNS descrito no algoritmo 4 da seção 3. Em seguida, os embarques heurísticos de passageiros das melhores soluções

obtidas nas iterações do algoritmo são removidos e calculados de forma exata, como descrito na seção 3.1. O procedimento retorna a melhor solução calculada.

5. Experimento Computacional

Por se tratar de um modelo pioneiro na literatura, o projeto de validação dos algoritmos de solução contemplou construir casos testes para o problema, buscar a solução exata das instâncias de teste, encontrar soluções exatas compostas e, finalmente, uma etapa de solução heurísticas compostas. A etapa exata foi realizada através do *solver* de programação matemática *Gurobi* na versão 7.5.2 [9], utilizando o modelo da seção 2. Para a maioria das instâncias, o *solver* foi incapaz de alcançar até mesmo um limite de solução considerado o horizonte de 80.000 segundos de execução. A heurística Lin-Kernighan foi implementada segundo a versão 2.0.8 do trabalho de Helsgaun [10].

Os experimentos foram executados em uma máquina Workstation, com processador Intel Core i7-2600K, 3.40GHz com 4GB de memória RAM e sistema operacional Ubuntu 16.04.3 LTS 64-bit. Os algoritmos do trabalho foram implementados na linguagem C++ na versão 11. O compilador utilizado foi o GNU C++ Compiler (g++) versão 5.4.0.

5.1 Constituição das Instâncias de Teste

Para sistematizar a produção de instâncias para o problema, a pesquisa propôs um gerador de instâncias. O gerador recebe dois parâmetros de entrada: a quantidade de localidades do grafo G , n , e a capacidade do carro, k . Na primeira fase de criação das instâncias, o gerador produz um grafo completo com arestas de custo sorteadas em distribuição uniforme entre 100 e 250. Se o valor da aresta é sorteado uma única vez, a instância é dita simétrica. Se é realizado um sorteio para cada direção da aresta, a instância é dita assimétrica.

A quantidade de passageiros em cada localidade i foi sorteada em distribuição uniforme no intervalo $[0, 1.5[k]]$. Cada passageiro tem seu destino sorteado aleatoriamente em sorteio uniforme sobre as localidades do grafo diferente daquela que o passageiro ocupa. A tarifa admissível para o passageiro é calculada pelo valor do caminho mais curto entre a localidade de hospedagem do passageiro e a localidade de destino, multiplicado por um valor aleatório sorteado uniformemente no intervalo $[2,3]$.

Os prêmios das localidades são sorteados em distribuição uniforme entre 100 a 250. B_{min} é fixado como 40% da soma de todos os prêmios das localidades. As penalidades por deixar de visitar uma localidade são sorteadas no intervalo $[1,30]$, considerando-se $b_1 = p_1 = 0$, a penalidade e o bônus da localidade inicial da rota.

As instâncias geradas como anteriormente descritas, são classificadas como instâncias aleatórias do Grupo A. A partir das instâncias do Grupo A são constituídas instâncias denominadas do Grupo B por um procedimento que altera a distribuição dos valores dos bônus da instância em função das penalidades da localidade. A redistribuição dos bônus

é realizada com o auxílio de uma roleta [11] em função das penalidades das localidades, atribuindo-se maior chance de prêmios maiores para localidades com penalidades maiores. A partir do Grupo B, forma-se o grupo C de instâncias através de uma alteração em relação a quantidade de passageiros de cada vértice i . Define-se a atratividade da localidade como a soma do prêmio b_i com a penalidade p_i . A quantidade de passageiros existentes em uma localidade i é tornada inversamente proporcional à atratividade dessa localidade. A redistribuição de passageiros é realizada em todas as localidades através de um sorteio em roleta, com maior chance de ocupação para os vértices com menor atratividade. Caso uma alteração na localidade de origem do passageiro faça coincidir destino com origem, é gerado uma nova localidade de destino para este passageiro. Por último, considerando que podem ocorrer alterações de origem, a tarifa admissível dos passageiros é recalculada para cada passageiro conforme o procedimento de caminho mais curto já descrito anteriormente.

5.2 Parametrização

Os parâmetros de inicialização do algoritmo foram definidos pela ferramenta IRACE (Iterated Racing for Automatic Algorithm Configuration) [12]. Para o banco de treinamento da ferramenta foram utilizadas 12 instâncias, 6 simétricas e 6 assimétricas, sorteadas aleatoriamente. As instâncias envolveram 10, 20, 30, 40, 50 e 100 localidades. Foram sorteadas quatro instâncias por grupo. A tabela 1 exibe a distribuição das instâncias. As instâncias de treinamento não são usadas no conjunto de instâncias do experimento.

Table 1. Instâncias para treinamento do IRACE

Inst.	n	k	Grupo	Inst.	n	k	Grupo
s10-3	10	3	A	a10-3	10	3	A
s20-4	20	4	B	a20-4	20	4	B
s30-4	30	4	B	a30-4	30	4	B
s40-4	40	4	C	a40-4	40	4	C
s50-4	50	4	A	a50-4	50	4	A
s100-4	100	4	C	a100-4	100	4	C

A Tabela 2 mostra os limites fornecidos ao IRACE para a parametrização. A Tabela 3 apresenta os valores retornados e que foram utilizados nos experimentos.

Table 2. Limites determinados para o IRACE

Parâmetro	Descrição	Intervalo
<i>IterMax</i>	Número de Iterações de HADeS 1 e 2	{50, 80, 100, 150, 200, 250, 300}
<i>IterSearchMax</i>	Número de Iterações do Proc. VNS	[5,20]
<i>GurobiTime</i>	Tempo limite em segundos para a execução do <i>solver Gurobi</i>	[30,60]
<i>FileSize</i>	Quantidade das melhores soluções	[2,10]
α	Parâmetro utilizado na LRC	[0,1]

Table 3. Valores obtidos pelo IRACE

Parâmetro	HERS	HERS2	GRASP
<i>IterMax</i>	250	300	300
<i>IterSearchMax</i>	-	17	20
<i>GurobiTime</i>	43	54	52
<i>FileSize</i>	9	10	7
α	-	-	0.42

O parâmetro *IterMax* define a quantidade máxima de iterações realizada por todos os algoritmos. O parâmetro *IterSearchMax* anota as iterações da busca local no VNS. O parâmetro *GurobiTime*, determina o tempo máximo, em segundos utilizado pelo *solver Gurobi* durante a determinação do embarque exato em uma rota. O parâmetro determina que o *solver* é interrompido quando alcançar o limite de tempo, retornando nesse caso, uma solução aproximada. Caso o *solver* retorne uma solução aproximada que não melhore a solução anteriormente existente, a solução do *solver* é descartada. O parâmetro *FileSize* determina a quantidade máxima das melhores soluções que o algoritmo deve guardar no decorrer da execução. Por fim, o parâmetro *Alpha* é utilizado pela LRC no algoritmo GRASP.

5.3 Resultados

Para a obtenção da solução ótima dos casos teste, as instâncias foram submetidas ao *solver Gurobi* segundo o modelo da seção 2 e executado até o limite de 80.000 segundos para cada instância. As heurísticas HERS1, HERS2, e a meta-heurística GRASP foram executadas 30 vezes em cada instância com os parâmetros definidos na tabela 3. As heurísticas HES/CCP e HES/PCV, como algoritmos determinísticos, foram executadas apenas uma vez.

Os valores retornados pelo *solver* foram: *Incumbent*, o resultado final encontrado para a instância, e *Best Bound*, o limite inferior calculado para a instância. *GAP* é a diferença em porcentagem dos valores *Incumbent* e *Best Bound*. Por fim, o tempo em segundos levado para executar o algoritmo. As tabelas 4 a 6 mostram os resultados obtidos pela execução da solução exata do modelo da seção 2.

O experimento comprovou que o método exato só conseguiu encontrar a solução para instâncias com apenas 10 vértices em todos os grupos. Para instâncias de tamanho 20, a execução do *solver* foi interrompida pelo tempo, encontrando soluções aproximadas com *GAP* entre 80% a 85%. Nas demais instâncias, o *solver* acabou ultrapassando a capacidade de memória RAM da máquina com 4Gb de RAM.

Table 4. Valores do Gurobi para o grupo A

Inst	Incumb	BestB	Gap	T(s)
s10-3	380,10	380,10	0%	8317,12
s10-4	365,38	365,38	0%	27996,79
s20-4	746,47	134,81	81,90%	80000
a10-3	302,53	302,53	0%	3152,56
a10-4	316,08	316,08	0%	12179,72
a20-4	672,4	104,47	84,50%	80000

Table 5. Valores do Gurobi para o grupo B

Inst	Incumb	BestB	Gap	T(s)
s10-3	386,78	386,78	0%	8864,55
s10-4	354,97	354,97	0%	15062,21
s20-4	729,07	112	84,60%	80000
a10-3	304,86	304,857	0%	3138,81
a10-4	314,35	314,35	0%	20435,62
a20-4	679,4	104,66	84,60%	80000

Table 6. Valores do Gurobi para o grupo C

Inst	Incumb	BestB	Gap	T(s)
s10-3	366,997	366,997	0%	7042,14
s10-4	344,84	344,84	0%	6649,28
s20-4	698,14	112,08	83,94%	80000
a10-3	305,64	305,64	0%	2582,97
a10-4	287,06	287,06	0%	8093,68
a20-4	671,07	104,91	84,40%	80000

Para os experimentos com os demais algoritmos os valores retornados foram: a solução mínima, média e tempo médio de execução dos algoritmos HERS 1, HERS 2, e GRASP. Por fim, a solução obtida e tempo de execução das heurísticas HES/PCV e HES/CPP.

Os resultados obtidos para cada grupo de instâncias, podem ser visualizados nas tabelas de 25 a 28.

Com o intuito de facilitar a avaliação dos resultados obtidos, a Tabela 7 demonstra a comparação entre vitórias e derrotas de cada algoritmo em relação a seus melhores resultados descritos nas tabelas de 25 a 28.

Table 7. Comparação dos melhores resultados de todos os grupos (Vitórias x Derrotas).

	HERS1	HERS2	GRASP	H/CPP	H/PCV
HERS1	-	10 x 36	6 x 40	53 x 1	54 x 0
HERS2	36 x 10	-	17 x 26	54 x 0	54 x 0
GRASP	40 x 6	26 x 17	-	54 x 0	54 x 0
H/CPP	1 x 53	0 x 54	0 x 54	-	39 x 15
H/PCV	0 x 54	0 x 54	0 x 54	15 x 39	-

Pelo resultado quantitativo alcançado na tabela 7, os algoritmos que obtiveram melhores performances relativas foram o HERS 2 e o GRASP, superando qualitativamente as heurísticas do tipo HES (CPP e PCV). O GRASP obteve um melhor resultado do que o HERS 2 com uma diferença de 9 instâncias.

5.4 Teste Estatístico

Para examinar as conclusões qualitativas com os resultados obtidos, realizou-se um teste estatístico post-hoc de Conover [4] aplicado ao teste de Friedman [6]. A tabela 8 exibe os p-valores retornados pelo teste de Friedman em ambos os grupos de instâncias. Nossa análise levou em consideração o nível de significância de 0,05.

As tabelas 9 a 11 resumem o teste de Post-hoc de Conover aplicado ao teste de Friedman nos grupos.

Os testes de Friedman e Post-hoc de Conover indicam que os algoritmos HERS1, HERS2, GRASP, e HES possuem

Table 8. p-valores do teste de Friedman

Grupo	p-valor
A	3,97E-13
B	4,77E-12
C	1,09E-11

Table 9. Teste post-hoc no grupo A

	HERS1	HERS2	GRASP	H/CPP
HERS2	4,50E-09	-	-	-
GRASP	5,30E-14	6,70E-03	-	-
H/CPP	1,20E-15	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	4,30E-03

Table 10. Teste post-hoc no grupo B

	HERS1	HERS2	GRASP	H/CPP
HERS2	4,10E-03	-	-	-
GRASP	2,40E-04	2,89E-02	-	-
H/CPP	2,00E-16	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	2,40E-04

Table 11. Teste post-hoc no grupo C

	HERS1	HERS2	GRASP	H/CPP
HERS2	8,40E-05	-	-	-
GRASP	2,20E-06	3,00E-01	-	-
H/CPP	3,80E-16	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	1,80E-01

diferença estatística em relação ao resultado obtido dos grupos de instâncias A e B em virtude de ambos os testes terem exibido p-valores menores do que o valor de significância 0,05. O teste do algoritmo GRASP e HERS2 mostrou-se inconclusivo para o grupo C.

Para examinar o desempenho em tempo dos algoritmos, foi realizado o teste estatístico de Friedman e Post-hoc de Conover em relação ao tempo médio de execução dos algoritmos HERS1, HERS2, e GRASP. Os algoritmos HERS são executados apenas uma vez por serem determinísticos. A tabela 12 exibe os p-valores retornados pelo teste de Friedman em ambos os grupos de instâncias, com o valor de significância de 0,05.

Table 12. p-valores do teste de Friedman para o tempo

Grupo	p-valor
A	6,76E-14
B	7,45E-14
C	3,22E-13

As Tabelas 13 a 15 resumem o teste post-hoc de Conover aplicado ao teste de Friedman em ambos os grupos. Em relação ao tempo médio de execução dos algoritmos, fica comprovado pelo teste de Friedman e post-hoc Conover que possuem diferença estatística.

5.4.1 Experimento com HERS2 x GRASP

No experimento descrito na seção anterior, os algoritmos que obtiveram melhores resultados foram o HERS2 e o GRASP. No grupo C, apesar do GRASP ter tido um melhor desem-

Table 13. Teste post-hoc relativo ao tempo no grupo A

	HERS1	HERS2	GRASP	H/CPP
HERS2	2,00E-16	-	-	-
GRASP	9,40E-13	2,30E-04	-	-
H/CPP	4,30E-12	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	3,90E-09

Table 14. Teste post-hoc relativo ao tempo no grupo B

	HERS1	HERS2	GRASP	H/CPP
HERS2	2,00E-16	-	-	-
GRASP	1,20E-12	1,20E-05	-	-
H/CPP	3,30E-10	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	2,00E-10

Table 15. Teste post-hoc relativo ao tempo no grupo C

	HERS1	HERS2	GRASP	H/CPP
HERS2	3,00E-16	-	-	-
GRASP	4,00E-12	2,00E-02	-	-
H/CPP	8,30E-11	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,00E-16	2,00E-07

penho qualitativo em 9 instâncias a mais do que o HERS 2, o teste post-hoc exibiu resultado inconclusivo. Para aprofundar a análise em busca de obter um indicador conclusivo, desenvolveu-se um teste com base no tempo médio de execução de cada algoritmo.

Para o experimento, utilizou-se o tempo médio de execução computado para cada instância do experimento inconclusivo, segundo o critério de parada de cada algoritmo. Com este propósito, ao invés do HERS2 e GRASP utilizarem o parâmetro *IterMax* (seção 5.2) como critério de parada sem melhorias da solução, o parâmetro utilizado nos algoritmos foi o tempo médio de execução do algoritmo rival para a instância em curso. Desta forma, testa-se a condição do algoritmo receber um tempo maior do tempo de parada projetado, e ser capaz de, usando o novo tempo, melhorar sua solução. Ou se, recebendo um tempo menor que o projetado, piora o seu desempenho qualitativo. Basicamente o teste explora a sensibilidade do algoritmo em relação à variação do tempo. A seguir pode ser visualizado os resultados para os demais grupos de instâncias pelas Tabelas de 29 a 31.

Com o intuito de facilitar a avaliação dos resultados obtidos, a Tabela 16 ilustra a comparação entre vitórias e derrotas de cada algoritmo em relação a seus melhores resultados descrito nas Tabelas 28 (HES) e 29 a 31.

Table 16. Comparação dos melhores resultados apresentados de todos os grupos no experimento HERS X GRASP (Vitórias x Derrotas).

	HERS2	GRASP	H/CPP	H/PCV
HERS2	-	25 x 18	54 x 0	54 x 0
GRASP	18 x 25	-	54 x 0	54 x 0
H/CPP	0 x 54	0 x 54	-	39 x 15
H/PCV	0 x 54	0 x 54	15 x 39	-

A tabela 17 ilustra os p-valores do teste de Friedman [6] e

as tabelas 18 a 20 o teste de post-hoc de Conover [4]. Ambos os testes estatísticos são em relação as melhores soluções dos algoritmos HERS2 e GRASP.

Table 17. p-valores do teste de Friedman para o teste HERS2 x GRASP

Grupo	p-valor
A	6,76E-14
B	7,45E-14
C	3,22E-13

Table 18. Teste post-hoc do experimento HERS2 x GRASP no grupo A

	HERS2	GRASP	H/CPP
GRASP	6,41E-03	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	8,80E-04

Table 19. Teste post-hoc do experimento HERS2 x GRASP no grupo B

	HERS2	GRASP	H/CPP
GRASP	1,30E-03	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	7,70E-07

Table 20. Teste post-hoc do experimento HERS2 x GRASP no grupo C

	HERS2	GRASP	H/CPP
GRASP	5,50E-01	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	1,50E-01

O teste de Friedman e o post-hoc de Conover ilustrado pelas tabelas 21 a 24 são em relação ao tempo médio de execução de cada algoritmo.

Table 21. p-valores do teste de Friedman para o teste HERS2 x GRASP relativo ao tempo

Grupo	p-valor
A	2,13E-10
B	1,85E-10
C	3,50E-10

Table 22. Teste post-hoc do experimento HERS2 x GRASP no grupo A relativo ao tempo

	HERS2	GRASP	H/CPP
GRASP	1,00E+00	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	4,20E-10

Como mostrado pela tabela 16, o algoritmo HERS2 obteve um melhor desempenho em relação ao GRASP, contabilizando um resultado superior em 7 instâncias. O teste estatístico de

Table 23. Teste post-hoc do experimento HERS2 x GRASP no grupo B relativo ao tempo

	HERS2	GRASP	H/CPP
GRASP	1,00E+00	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	6,30E-11

Table 24. Teste post-hoc do experimento HERS2 x GRASP no grupo C relativo ao tempo

	HERS2	GRASP	H/CPP
GRASP	1,00E+00	-	-
H/CPP	2,00E-16	2,00E-16	-
H/PCV	2,00E-16	2,00E-16	2,80E-08

Friedman e post-hoc Conover, indicaram que os algoritmos GRASP e HERS2 possuem diferença estatística nos grupos A e B em relação às melhores soluções obtidas. Como a heurística HERS2 alcança as melhores soluções nesse grupo, pode ser considerada conclusivamente superior em desempenho nesses dois grupos de instâncias. Porém, em relação ao grupo C de instâncias, o teste apresenta resultado inconclusivo.

O teste estatístico aplicado ao tempo médio de execução dos algoritmos é inconclusivo para os três grupos de instâncias, neutralizando a consideração desse parâmetro para a decisão de melhor desempenho.

6. Considerações Finais

Este trabalho definiu e modelou um novo problema de roteamento denominado Problema do Caixeiro Viajante com Coleta de Prêmios e Passageiros (PCV-CPP). O problema aborda a otimização do espaço ocioso em carros que transportam encomendas e, conseqüentemente, podem trafegar com espaços ociosos entre localidades.

A aplicação prática tem potencial para produzir impactos positivos tanto econômicos, como ambientais e sociais. O problema descrito compartilha características com o clássico PCV-CP, contudo os resultados do experimento computacional sugerem fortemente que o novo modelo possui uma complexidade significativamente superior ao PCV-CP e que, nas instâncias testadas, possui soluções significativamente afastadas da composição de mínimos locais, considerados os mínimos associados à solução dos problemas que compõem o modelo, a saber, o caixeiro viajante com coleta de prêmios e o problema do embarque ótimo de passageiros.

Paralelamente ao exame desse novo modelo, o trabalho desenvolveu a proposta de uma nova classe de heurísticas construtivas, presentemente denominada de HERS. O experimento computacional desenvolvido permitiu sustentar que as heurísticas HERS possuem potencial para alcançar boas soluções para o problema examinado, tanto quando os resultados exatos disponíveis são confrontados com as soluções heurísticas, como na comparação de uma heurística GRASP desenvolvida para o problema. Observe-se que a heurística

GRASP emprega em sua etapa de busca local, estratégias de buscas e vizinhanças semelhantes às empregadas nas etapas das HERS, diferenciando-se essencialmente no processo de decomposição da etapa construtiva. As heurísticas HERS permitem, conforme a estrutura de decisão do problema, decompor decisões acopladas em etapas independentes, permitindo a aplicação de diferentes estratégias de solução a cada etapa dessa decomposição. A nova classe de heurísticas proposta, possui formulação simples e pode ser considerada uma forma de generalização do método de decomposição proposto pela meta-heurística GRASP, sendo aplicável para a solução de problemas que possam ter sua solução final decomposta em decisões parciais e sequenciais. A lógica embutida no procedimento é a construção de uma solução global através da combinação de boas soluções parciais do problema.

Como trabalho futuro permanece em aberto a aplicação das heurísticas HERS na solução de outros tipos de problemas e o desenvolvimento de outras meta-heurísticas para a solução do PCV-CPP.

Agradecimentos

Este trabalho foi suportado pelo CNPq através dos processos 302387/2016-1 e 306702/2017-7.

Contribuições dos autores

Os autores contribuíram igualmente em todas as fases da confecção do trabalho.

Referências

- [1] BALAS, E. The prize collecting traveling salesman problem. *Networks* 19, 6 (1989), 621–636.
- [2] BASTOS, R. E. M., GOLDBARG, M. C., AND GOLDBARG, E. F. G. O problema do caixeiro viajante com passageiros e lotação. In *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional* (2016).
- [3] CALHEIROS, Z. S. A. O problema do caixeiro viajante com passageiros. Master's thesis, Universidade Federal do Rio Grande do Norte, 2017.
- [4] CONOVER, W. J. Statistics of the kolmogorov-smirnov type. *Practical Nonparametric Statistics* 8, 2 (1999), 428–473.
- [5] FEO, T. A., AND RESENDE, M. G. C. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8, 2 (1989), 67–71.
- [6] FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32, 200 (1937), 675–701.
- [7] GOLDBARG, M. C., GOLDBARG, E. F. G., ASCONAVIETA, P. H., MENEZES, M. S., AND LUNA,

- H. P. A transgenetic algorithm applied to the traveling car renter problem. *Expert Systems with Applications* 40, 16 (2013), 6298–6310.
- [8] GRIBKOVSKAIA, I., LAPORTE, G., AND SHYSHOU, A. The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research* 35, 9 (2008), 2908–2924.
- [9] GUROBI, O. Gurobi optimizer reference manual. <https://www.gurobi.com> (2019).
- [10] HELSGAUN, K. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research* 126, 1 (2000), 106–130.
- [11] LIPOWSKI, A., AND LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications* 391, 6 (2012), 2193–2196.
- [12] LÓPEZ-IBÁÑEZ, M., DUBOIS-LACOSTE, J., CÁCERES, L. P., BIRATTARI, M., AND STÜTZLE, T. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3 (2016), 43–58.
- [13] MILLER, C. E., TUCKER, A. W., AND ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)* 7, 4 (1960), 326–329.
- [14] MLADENOVIĆ, N., AND HANSEN, P. Variable neighborhood search. *Computers & Operations Research* 24, 11 (1997), 1097–1100.
- [15] NOURINEJAD, M. *Dynamic optimization models for ridesharing and carsharing*. PhD thesis, Department of Civil Engineering, University of Toronto, 2014.
- [16] SABRY, G. A., GOLDBARG, M. C., AND GOLDBARG, E. F. G. Problema do caixeiro viajante alugador com passageiros: Uma abordagem algorítmica. In *Proceedings of the XVIII CLAIO, the Latin-Iberoamerican Conference on Operations Research* (2016), vol. 1, Ediciones UC, pp. 764–771.
- [17] SILVA, J. G. S. Algoritmos de solução para o problema do caixeiro viajante com passageiros e quota. Master’s thesis, Universidade Federal do Rio Grande do Norte, 2017.

Table 25. Resultado do experimento computacional do grupo A

Inst.	HERS1			HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	383,69	385,93	1,38	383,69	383,79	1,64	383,69	384,61	1,44
s10-4	380,02	380,73	1,74	372,30	378,02	2,40	372,30	378,24	2,39
s20-4	494,70	3511,56	33,98	479,35	493,19	6,29	478,77	497,38	5,53
s30-4	638,52	3660,77	39,83	606,82	635,53	16,44	586,21	623,62	13,60
s40-4	817,84	3853,19	314,73	762,13	799,17	34,39	779,07	800,98	30,75
s50-4	897,54	3920,56	324,23	899,20	918,11	51,56	876,87	913,22	45,47
s100-4	1950,81	32076,07	394,24	1855,30	1988,59	176,21	1779,05	1938,49	188,67
s150-4	2755,42	33058,13	3306,18	2665,29	2944,67	569,21	2529,91	2873,58	566,04
s200-4	3499,87	33812,47	3608,62	3426,66	3756,07	997,10	3435,49	3878,58	986,33
a10-3	302,53	3302,53	32,32	302,53	302,53	2,42	302,53	302,53	2,44
a10-4	320,09	3333,94	31,61	320,09	320,09	2,25	320,09	322,14	1,76
a20-4	422,12	3437,27	311,57	393,91	410,84	13,58	391,47	411,81	10,30
a30-4	593,02	3628,58	320,03	579,31	608,17	33,05	591,02	611,28	21,47
a40-4	795,43	3823,94	342,23	758,73	805,89	70,90	744,58	809,97	49,72
a50-4	931,26	3959,35	351,92	910,39	947,82	86,24	892,71	941,26	75,64
a100-4	1798,72	31886,03	3237,93	1789,95	1884,02	335,28	1775,34	1881,34	356,09
a150-4	2772,34	33206,64	3638,38	2708,49	3044,29	905,22	2720,56	2910,86	983,82
a200-4	3683,88	33954,64	31174,30	3675,32	3970,16	1715,61	3656,62	4016,24	1783,50

Table 26. Resultado do experimento computacional do grupo B

Inst.	HERS1			HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	386,78	386,85	1,49	86,78	386,78	1,59	386,78	386,78	1,61
s10-4	376,05	376,05	1,67	358,82	360,99	2,67	358,82	360,24	2,21
s20-4	472,36	488,49	3,62	466,87	481,89	5,41	471,52	485,23	4,62
s30-4	586,17	612,82	7,36	589,64	608,77	14,73	578,98	612,01	11,53
s40-4	760,14	777,96	14,84	748,93	772,50	34,70	749,11	783,68	30,39
s50-4	823,62	840,76	21,985	824,93	840,96	51,05	813,41	845,00	39,85
s100-4	1748,40	1882,41	76,40	1682,23	1855,22	186,68	1718,54	1878,70	169,00
s150-4	2433,18	2793,62	292,13	2433,47	2723,76	533,26	2367,07	2646,50	652,15
s200-4	3160,94	3400,98	613,09	3122,77	3500,20	1059,96	3186,02	3540,71	1033,74
a10-3	310,62	310,62	4,03	310,62	310,77	4,15	310,62	310,62	4,12
a10-4	323,17	323,17	1,46	314,35	321,08	1,78	320,38	321,03	1,55
a20-4	412,24	419,37	9,28	407,78	418,33	11,67	407,78	417,45	9,57
a30-4	601,64	619,22	24,18	597,40	612,28	28,31	592,36	614,89	22,92
a40-4	748,34	776,05	42,10	723,40	754,51	57,83	700,81	745,74	51,85
a50-4	874,03	909,23	55,83	869,61	903,02	93,48	867,21	925,32	77,03
a100-4	1560,82	1637,06	226,23	1576,20	1671,85	358,10	1580,84	1671,22	357,80
a150-4	2416,81	3090,87	679,98	2443,50	2837,24	935,53	2368,83	2765,67	992,18
a200-4	3132,14	3330,79	1170,62	3133,13	3418,32	1540,30	3256,03	3531,54	1672,08

Table 27. Resultado do experimento computacional do grupo C

Inst.	HERS1			HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	388,93	389,67	1,61	371,66	385,58	1,81	376,48	386,84	1,58
s10-4	344,84	351,76	1,81	344,84	346,59	2,44	344,84	352,52	1,88
s20-4	572,30	587,92	4,02	467,81	488,67	6,77	469,43	496,26	5,65
s30-4	615,69	627,47	6,69	597,63	623,84	13,61	589,43	630,65	11,93
s40-4	763,20	807,50	16,96	753,71	797,20	34,06	751,52	802,17	33,64
s50-4	881,61	905,27	21,04	865,80	898,20	51,72	860,59	897,76	57,37
s100-4	2104,89	2347,22	83,16	1871,72	2081,42	176,51	1831,02	2042,30	208,59
s150-4	3293,19	4337,61	156,22	2559,10	3515,56	433,49	2569,65	3095,70	542,65
s200-4	6325,40	6980,42	244,75	3973,02	5366,69	663,65	3714,89	4602,77	1019,45
a10-3	310,50	346,63	4,35	305,64	305,80	4,97	305,64	306,12	5,14
a10-4	287,06	287,06	1,43	287,06	287,06	1,76	287,06	287,06	1,53
a20-4	411,56	418,61	9,40	411,56	417,32	12,51	414,48	422,33	9,74
a30-4	602,53	677,70	22,78	595,09	624,48	37,37	597,26	626,50	20,64
a40-4	742,79	828,83	44,20	747,21	796,42	61,28	742,79	803,37	44,97
a50-4	859,33	897,96	57,15	859,65	893,06	82,55	867,01	929,80	59,04
a100-4	1679,63	1964,05	195,17	1806,01	1904,55	337,61	1732,51	1866,01	314,92
a150-4	3279,70	4213,97	507,13	2952,09	3440,97	785,87	2675,73	3140,16	921,81
a200-4	6146,43	6907,04	896,90	4171,12	5858,10	1315,09	3681,24	4794,35	1757,06

Table 28. Resultados da heurística HES

Inst.	Grupo A				Grupo B				Grupo C			
	CPP	T(s)	PCV	T(s)	CPP	T(s)	PCV	T(s)	CPP	T(s)	PCV	T(s)
s10-3	408,89	0,06	446,75	0,01	399,26	0,04	446,75	0,01	438,43	0,03	457,50	0,02
s10-4	452,92	0,02	414,27	0,02	429,45	0,02	414,27	0,01	464,29	0,01	424,48	0,02
s20-4	816,37	0,12	656,15	0,03	583,04	0,15	656,15	0,03	974,62	0,15	591,08	0,04
s30-4	696,70	0,30	911,85	0,12	658,35	0,19	911,85	0,12	898,22	0,15	804,15	0,12
s40-4	1029,23	3,28	994,60	0,29	924,86	1,48	994,60	0,25	1010,43	1,25	1139,00	0,27
s50-4	952,05	1,32	1253,45	0,43	950,43	1,40	1253,45	0,36	1020,04	1,31	1276,42	0,41
s100-4	2191,44	20,42	2936,37	1,43	2585,76	8,77	2936,37	1,43	2478,16	8,83	3309,13	1,38
s150-4	3062,84	103,93	8508,18	3,25	3669,69	301,87	8508,18	3,29	3538,67	301,63	8185,00	3,62
s200-4	3574,58	47,61	10077,50	6,36	4845,76	61,18	10077,50	6,37	7494,28	71,33	13965,20	6,39
a10-3	450,79	0,01	450,08	0,01	454,01	0,01	450,08	0,01	514,01	0,01	556,42	0,01
a10-4	371,70	0,03	352,13	0,01	505,68	0,02	352,13	0,01	363,68	0,02	346,78	0,01
a20-4	477,16	0,11	554,90	0,03	471,45	0,09	554,90	0,03	471,45	0,08	590,40	0,04
a30-4	789,88	0,16	828,32	0,07	766,64	0,07	828,32	0,05	983,56	0,07	837,98	0,07
a40-4	948,61	0,86	1123,78	0,13	1054,75	2,15	1123,78	0,10	1139,33	2,35	1083,40	0,11
a50-4	1054,63	0,51	1246,92	0,29	1092,68	0,49	1246,92	0,20	1114,34	0,43	1444,88	0,27
a100-4	2410,86	4,81	3308,43	0,83	3008,21	2,81	3308,43	0,83	2023,53	2,93	3074,80	0,84
a150-4	3393,90	5,29	6415,95	1,36	4033,57	4,95	6415,95	1,31	5888,26	4,83	4745,63	1,44
a200-4	7167,75	8,35	11257,80	2,92	7678,47	12,78	11257,80	2,96	5577,65	10,85	12070,50	2,85

Table 29. Resultado do experimento computacional comparando HERS2 e GRASP no grupo A

Inst.	HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	383,69	384,40	1,09	383,69	384,91	1,07
s10-4	372,30	377,98	2,07	372,30	378,35	2,05
s20-4	479,47	497,40	5,12	459,11	496,28	6,09
s30-4	602,78	635,41	13,24	606,90	629,69	16,16
s40-4	767,88	801,83	30,50	758,86	801,34	34,34
s50-4	890,07	917,15	45,62	863,42	913,24	51,54
s100-4	1797,67	2002,81	215,84	1822,27	1955,08	193,35
s150-4	2615,46	2968,05	749,08	2643,10	2976,07	725,76
s200-4	3447,27	3886,50	1414,90	3551,19	3957,08	1269,75
a10-3	302,53	302,53	2,09	302,53	302,53	2,07
a10-4	320,09	322,76	1,10	320,09	322,14	2,06
a20-4	391,47	413,19	10,21	391,47	411,81	13,16
a30-4	583,69	606,56	21,46	591,02	611,28	33,36
a40-4	782,30	806,41	49,99	744,58	809,97	70,60
a50-4	872,71	943,50	76,18	892,71	941,26	86,98
a100-4	1753,75	1886,23	400,82	1775,34	1881,34	360,46
a150-4	2826,13	3044,72	1226,99	2720,56	2910,86	1054,93
a200-4	3625,82	3924,76	2270,03	3656,62	4016,24	2005,87

Table 30. Resultado do experimento computacional comparando HERS2 e GRASP no grupo B

Inst.	HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	386,78	386,78	1,06	386,78	386,78	1,04
s10-4	358,82	361,03	2,06	358,82	360,15	2,05
s20-4	469,11	484,10	4,10	471,10	484,06	5,08
s30-4	585,99	607,30	11,23	591,49	615,62	14,17
s40-4	756,11	775,53	30,47	749,25	782,77	34,30
s50-4	821,91	839,88	39,54	821,38	846,06	51,51
s100-4	1675,81	1856,69	186,89	1734,03	1860,73	200,45
s150-4	2423,32	2707,75	852,18	2409,60	2621,39	664,86
s200-4	3106,69	3529,54	1465,63	3178,63	3625,43	1329,83
a10-3	310,62	310,62	4,12	310,62	310,62	4,09
a10-4	314,35	321,76	1,10	320,38	321,03	1,05
a20-4	412,10	419,00	9,28	407,78	417,45	11,16
a30-4	592,05	611,41	22,46	592,36	614,89	28,35
a40-4	713,27	752,03	51,97	700,81	745,74	57,68
a50-4	866,47	903,82	78,18	867,21	925,32	93,90
a100-4	1562,27	1679,35	404,10	1580,84	1671,22	395,27
a150-4	2487,59	2853,72	1241,15	2368,83	2765,67	1085,10
a200-4	3110,52	3389,16	2159,41	3256,03	3531,54	1838,32

Table 31. Resultado do experimento computacional comparando HERS2 e GRASP no grupo C

Inst.	HERS2			GRASP		
	Min	Méd.	Tm (s)	Min	Méd.	Tm (s)
s10-3	367,00	385,32	1,09	376,48	386,99	1,06
s10-4	344,84	350,54	1,08	344,84	351,86	2,05
s20-4	467,58	488,29	5,12	469,43	498,01	6,10
s30-4	617,24	630,63	11,21	600,37	627,41	13,17
s40-4	753,59	794,07	33,56	758,75	806,49	34,69
s50-4	881,63	902,32	57,66	858,13	906,93	51,53
s100-4	1841,88	2043,55	220,85	1732,53	2014,02	192,25
s150-4	2652,65	3452,28	597,59	2609,29	3151,22	509,06
s200-4	3780,92	5055,44	1128,18	3628,96	4497,03	813,74
a10-3	305,64	305,64	5,11	305,64	306,12	4,08
a10-4	287,06	287,06	1,10	287,06	287,06	1,07
a20-4	411,56	417,66	9,21	414,48	422,33	12,16
a30-4	585,28	620,18	20,57	597,26	626,50	37,38
a40-4	739,72	797,31	44,87	742,79	803,37	61,66
a50-4	854,03	892,82	60,29	867,01	929,80	82,90
a100-4	1704,92	1866,16	334,29	1732,51	1866,01	349,69
a150-4	2842,33	3395,07	980,73	2675,73	3140,16	861,44
a200-4	4425,73	5647,04	1886,81	3681,24	4794,35	1490,68