UMA ABORDAGEM ILS PARA TRATAR O PROBLEMA DO CAIXEIRO VIAJANTE PRETO E BRANCO

Paola Pinto Cazetta
Departamento de Informática - Universidade Federal de Viçosa
Campus da UFV, 36.570-900, Viçosa - MG - Brasil
pcazetta@gmail.com

Luciana Brugiolo Gonçalves
Departamento de Informática - Universidade Federal de Viçosa
Campus da UFV, 36.570-900, Viçosa - MG - Brasil
lbrugiolo@ufv.br

RESUMO

O Problema do Caixeiro Viajante Preto e Branco (PCV-PB) é definido sobre um grafo em que os vértices são particionados em vértices pretos e brancos. O objetivo do problema é encontrar um ciclo hamiltoniano de custo mínimo, como no PCV, mas considerando restrições adicionais relacionadas à cardinalidade e ao comprimento. A restrição de cardinalidade obriga que o número de vértices brancos entre dois vértices pretos esteja sujeito a um limite, enquanto o comprimento está relacionado à distância máxima entre dois vértices pretos. Uma abordagem ILS é utilizada para tratar o problema. Para avaliar a estratégia proposta foi gerado um conjunto de instâncias. Os resultados do algoritmo proposto foram comparados com os obtidos por uma estratégia da literatura baseada em GRASP e VND. Nestes resultados é possível observar que o ILS é superior tanto em relação à qualidade da solução quanto em relação ao tempo de CPU gasto.

PALAVARAS CHAVE. Caixeiro Viajante Preto e Branco, Heurística, ILS.

Áreas principais: Metaheurística, Otimização Combinatória, Logística e Transporte.

ABSTRACT

The Black and White Traveling Salesman Problem (TSP-BW) is defined on a graph in which vertices are partitioned into black and white vertices. Similar to the TSP, the objective of the TSP-BW is to find a Hamiltonian cycle of minimum cost; however, TSP-BW considerate additional constraints related to cardinality and length. The cardinality constraint limits the number of white vertices between two black vertices, while the length constraint is related to the maximum distance between two black vertices. An ILS approach was used to implement a solution for the TSP-BW problem. Additionally, the proposed strategy was evaluated using a generated set of instances. The obtained results were compared with results previously obtained in a related work, which strategy is based on GRASP and VND. The comparison shows that an ILS approach is superior in both terms of quality and CPU time.

KEYWORDS. Black and White Traveling Salesman, Heuristic, ILS.

Main Area: Metaheuristics, Combinatorial Optimization, Logistics and Transport.

1. Introdução

O clássico Problema do Caixeiro Viajante (PCV) consiste em encontrar o menor caminho para visitar uma série de cidades, onde cada uma destas deve ser visitada uma única vez, e retornar à cidade origem (Lopes *et al*., 2013; Ramos *et al*., *2003*).

Na literatura encontram-se diversas variações do PCV e uma destas é o Problema do Caixeiro Viajante Preto e Branco (PCV-PB), tema deste trabalho. O PCV-PB é definido sobre um grafo no qual os vértices que representam as cidades são classificados como pretos ou brancos. Além das restrições associadas ao PCV, no PCV-PB duas outras restrições são consideradas: uma associada a cardinalidade e outra ao comprimento. A restrição de cardinalidade está relacionada ao número máximo (Q) de vértices brancos entre dois pretos, e a restrição de comprimento define à distância máxima (L) que pode existir entre dois vértices pretos. Quando $Q = L = \infty$, o PCV-PB se reduz ao clássico PCV, um problema reconhecidamente NP-Difícil (Garey e Johnson, 1979), portanto o PCV-PB trata-se de uma generalização do PCV, sendo classificado como um problema NP-Difícil (Bourgeois *et al*., 2003).

Considerando-se um grafo direcionado, uma aplicação para o PCV-PB pode ser observada em agendamento de companhias aéreas. Durante o trajeto de um avião, é preciso realizar algumas operações como paradas para manutenção da aeronave e paradas relacionadas ao embarque e desembarque de passageiros. Considerando que há um número maior de embarque e desembarque do que etapas de manutenção do avião, as estações de embarque e desembarque podem ser associadas aos vértices brancos e os pontos de manutenção podem ser associados aos vértices pretos do PCV-PB. O objetivo é determinar uma sequência de voos que minimize o custo das operações aéreas de forma que a distância entre as estações de manutenção e a quantidade de operações de embarques e desembarques entre duas destas estações não ultrapassem um determinado limite (Talluri, 1998; Mak e Boland, 2000).

Já considerando um grafo não direcionado, o PCV-PB possui aplicações em telecomunicações. Utilizando uma rede de fibra ótica, por exemplo, há dois ambientes que podem ser definidos como centros de distribuição de rede e estações de escritório que necessitam do acesso à rede. Nesta aplicação é importante estabelecer uma distância entre esses ambientes, considerando que há mais escritórios do que pontos de distribuição. Os centros ou dispositivos de distribuição de rede são associados aos vértices pretos e as estações são associados aos vértices brancos. O objetivo é encontrar um ciclo de menor distância para que a estrutura forneça o serviço de rede com boa qualidade (Cosares *et al*., 1995; Wasem, 1991).

O estudo sobre estratégias para tratar o PCV-PB iniciou em Bourgeois *et al*. (2003). Para tratar o PCV-PB em grafos com no máximo 200 vértices o mesmo utiliza heurísticas construtivas e de busca local. O trabalho de Maciel *et al*. (2005) aplicou heurísticas baseadas em *Greedy Randomized Adaptative Search Procedure* (GRASP), *Variable Neighborhood Search* (VNS) e *Variable Neighborhood Descent* (VND) para tratar o problema em grafos com no máximo 200 vértices. Em Ghiani *et al*. (2006) os autores determinaram a solução ótima para instâncias com até 100 vértices por meio de uma estratégia *Branch-and-cut*. Já no trabalho de Jiang e*t al*. (2007) foi proposto um modelo de programação linear derivada de *Gavish-Grave LP* para o PCV-PB direcionado composta de $3n^2 + 2n$ restrições. E por último, em Bhattacharya *et al*. (2007) utiliza-se de um algoritmo aproximado que pode obter soluções com custo $\left(4 - \frac{3}{2k}\right)$ vezes próximo do custo da solução ótima quando um valor máximo de vértices brancos aparece entre dois vértices pretos consecutivos.

Neste trabalho é proposta uma abordagem heurística baseada na metaheurística *Iterated Local Search* (ILS) apresentada por Lourenço *et al*. (2010). Os resultados encontrados com a heurística proposta são comparados com os obtidos por Maciel *et al*. (2005), dado que esse trabalho é o que apresenta as estratégias heurísticas mais elaboradas para o problema em questão.

O trabalho está estruturado da seguinte forma: na Seção 2 o PCV-PB é definido e seu modelo matemático é apresentado. Na Seção 3 é apresentado o referencial teórico e, em sequência, a heurística proposta é detalhada na Seção 4. Por último, seguem as seções contendo os resultados e conclusões.

2. O Problema do Caixeiro Viajante Preto e Branco

Segundo Bourgeois *et al*. (2003), o PCV-PB pode ser definido sobre um grafo direcionado $G = (V, A)$ ou sobre um grafo não direcionado $G = (V, E)$, onde $V = \{v_o, v_1, v_2,...,v_{n-1}\}$ representa um conjunto de vértices, $A = \{(v_i, v_j): v_i \text{ e } v_j \in V, i \neq j\}$ um conjunto de arcos e $E = \{(v_i, v_j): v_i \text{ e } v_j \in V, i < j\}$ um conjunto de arestas de G. O conjunto de vértices V é dividido em dois subconjuntos disjuntos onde $V = B \cup P$. O subconjunto B contém os vértices classificados como brancos, enquanto P contém os vértices classificados como pretos, sendo que $|P| \geq 2$. Para o caso de um grafo G não direcionado, as notações (v_i, v_j) e (v_j, v_i) simbolizam a mesma aresta. Um valor $d_{ij} > 0$ é associado à aresta (v_i, v_j), representando a distância entre os vértices v_i e v_j.

No PCV-PB busca-se definir um circuito ou um ciclo hamiltoniano de comprimento mínima em G, que respeite as restrições do clássico PCV e também as restrições adicionais referentes ao PCV-PB. Circuito ou ciclo hamiltoniano é um caminho que inicia e termina em um mesmo vértice e que percorre todos os demais vértices do grafo sem repeti-los. Como descrito em Bourgeois *et al*. (2003), estas restrições adicionais são definidas da seguinte forma:

- Restrição de Cardinalidade: a quantidade de vértices brancos entre dois vértices pretos consecutivos não pode ser superior a um número inteiro positivo definido como Q.
- Restrição de Comprimento: a distância entre dois vértices pretos consecutivos não pode ultrapassar um número inteiro positivo L.

Para verificar se uma instância do problema admite soluções viáveis, duas restrições podem ser avaliadas:

i. $|B| > Q.|P|$: Se o número de vértices brancos for maior do que o número de vértices pretos multiplicado por Q, então a restrição de cardinalidade não pode ser satisfeita.
ii. $c_{i,j_1(i)} + c_{i,j_2(i)} > L$, para todo $v_i \in B$, onde $v_{j_1(i)}$ e $v_{j_2(i)}$ são os dois vértices pretos mais próximos de v_i. Isso significa que não existe um caminho que passe por v_i que seja capaz de satisfazer a restrição de comprimento L (Bourgeois *et al*., 2003).

Para uma definição formal, na próxima seção é apresentado um modelo de programação linear inteira que descreve o PCV-PB.

2.1 Modelo Matemático

Maciel *et al*. (2005) apresenta um modelo de programação linear inteira mista para o PCV-PB assimétrico. Para compreender o modelo, algumas definições são colocadas inicialmente: os vértices pertencentes ao conjunto V são numerados de 0 a n, considerando que a origem é o vértice 0. Um arco $(i, j) \in A$ tem como origem o vértice $i \in V$ e o vértice $j \in V$ como vértice de chegada. O conceito de cadeia neste trabalho se define como o conjunto de vértices e arestas entre dois vértices pretos consecutivos, que também estão incluídos na cadeia. Um conjunto de identificadores associados a cada uma das cadeias é representado por Φ, significando que as arestas pertencentes a este caminho devem estar associadas a um mesmo identificador. O modelo considera as variáveis binárias z_{ij} e x^k_{ij}, sendo que z_{ij} assume valor igual a 1 quando a aresta (i, j) de custo c_{ij} pertence à solução e assume valor 0, caso contrário. A variável x^k_{ij} assume valor igual a 1 se a aresta (i, j) de custo c_{ij} e identificador k pertence a solução e assume valor 0, caso contrário.

Como visto em Maciel *et al*. (2005), o PCV-PB pode ser modelado pelas equações de (1) a (12). A função objetivo (1) busca minimizar o custo total da rota. O conjunto de restrições (2) garante que somente uma aresta de um único identificador parte do vértice $i \in V$. Em (3) garante-se que somente uma aresta de um único identificador chegue ao vértice $i \in V$. As restrições (4) garantem que a aresta que chega a j e a que parte deste vértice sejam do mesmo identificador, para todo vértice branco ($j \in B$). Cada identificador $k \in \Phi$ caracteriza uma cadeia e, portanto, o conjunto de restrições (5) relaciona-se ao somatório do custo das arestas de cada cadeia de identificador $k \in \Phi$, cujo custo total deve ser menor ou igual ao comprimento limite L. Em (6), as restrições indicam que a soma do número de vértices brancos pertencentes a uma dada cadeia deve ser menor ou igual à cardinalidade Q. O conjunto de restrições (7) garante que o vértice origem da cadeia de identificador k se inicie com um vértice preto. E, de forma semelhante, as restrições em (8) garantem que o vértice destino pertença ao conjunto de vértices pretos. As restrições (9), (10) e (11) estão relacionados às restrições de fluxo e à formação de um ciclo hamiltoniano incluindo a origem. O conjunto de restrições (9) garante a não existência de sub-rotas e (10) obriga que, se $y_{ij} > 0$ para algum arco $(i, j) \in A$, então $z_{ij} = 1$. E, por último, as restrições (11) indicam que uma aresta presente na solução pode pertencer a apenas uma cadeia, assegurando a formação de uma única sub-rota contendo a origem. Seguindo a formulação, a solução obtida percorrerá todos os vértices uma única vez atendendo as restrições de cardinalidade e comprimento.

$$Min \sum_{(i,j)\in A} c_{ij}.z_{ij} \tag{1}$$

Sujeito a:

$$\sum_{k\in\Phi}\sum_{j\in V/\{i\}} x_{ij}^k = 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{k\in\Phi}\sum_{j\in V/\{i\}} x_{ji}^k = 1 \qquad \forall i \in V \tag{3}$$

$$\sum_{i\in V/\{j\}} x_{ij}^k = \sum_{l\in V/\{j\}} x_{jl,}^k \qquad \forall j \in B,\ \forall k \in \Phi \tag{4}$$

$$\sum_{(i,j)\in A} c_{ij}.x_{ij}^k \le L, \qquad \forall k \in \Phi \tag{5}$$

$$\sum_{(i,j)\in A} x_{ij}^k \le Q + 1, \qquad \forall k \in \Phi \tag{6}$$

$$\sum_{i\in P\ e\ (i,j)\in A} x_{ij}^k = 1, \qquad \forall k \in \Phi \tag{7}$$

$$\sum_{l\in P\ e\ (j,l)\in A} x_{jl}^k = 1, \qquad \forall k \in \Phi \tag{8}$$

$$\sum_{i=0}^{n} y_{ij} - \sum_{i=0}^{n} y_{ji} = 1, \qquad j = 1,2,\dots,n \tag{9}$$

$$y_{ij} \le n.z_{ij}, \qquad \forall (i,j) \in A \tag{10}$$

$$\sum_{k\in\Phi} x_{ij}^k = z_{ij,} \qquad \forall (i,j) \in A \tag{11}$$

$$x_{ij}^k, z_{ij} \in \{0,1\}, y_{ij} \ge 0, \qquad \forall (i,j) \in A \tag{12}$$

3. Referencial Teórico

O PCV-PB foi apresentado por Bourgeois *et al*. (2003), que para tratá-lo propuseram três heurísticas de construção, uma heurística de viabilização e uma heurística de melhoramento.

Com o mesmo objetivo, Maciel *et al*. (2005) desenvolveram estratégias heurísticas, consideradas mais bem sucedidas do que as propostas em Bourgeois *et al*. (2003), para tratar o PCV-PB. Dessa forma, a abordagem proposta neste trabalho utiliza como base algumas dessas heurísticas. Nas próximas subseções são detalhadas a heurística de construção C3_R, a estratégia de viabilização da solução e as buscas locais BL1, BL2, BL3, BL_2-opt e BL4 de (Maciel *et al*., 2005), utilizadas no ILS proposto.

No trabalho descrito em Ghiani *et al*. (2006) é proposta uma formulação linear inteira para o PCV-PB não direcionado e uma classe de inequações válidas. Um algoritmo *Branch-and-Cut* foi utilizado com o objetivo de resolver o problema. Diante dos resultados computacionais, verificou-se que o algoritmo obteve sucesso para instâncias do problema contendo no máximo 100 vértices.

Já em Jiang e*t al*. (2007), foi proposto um modelo de programação linear derivada de *Gavish-Grave LP* para o PCV-PB direcionado. O modelo é composto por $3n^2 + 2n$ restrições, e foi obtido por meio da redução do problema abordado para o clássico Problema do Caixeiro Viajante assimétrico, e adicionando n^2 restrições para tratar a cardinalidade e n^2 restrições para tratar o comprimento.

Por último, o trabalho de Bhattacharya *et al*. (2007) mostra um algoritmo aproximado onde o custo da solução do PCV-PB pode ser aproximado $\left(4-\frac{3}{2k}\right)$ vezes do custo da solução ótima quando um valor máximo de vértices brancos aparecem entre dois vértices pretos consecutivos. Além disso, quando exatamente k vértices brancos aparecem em uma cadeia, o limite de aproximação pode ser ligeiramente melhorado para $\left(4-\frac{15}{8k}\right)$. O limite de aproximação pode ser melhorado para 2.5 quando k assume valor 2.

3.1 Heurística de Construção C3_R

No trabalho de Maciel *et al*. (2005) foi proposto um algoritmo construtivo chamado de C3_R com característica randômica e adaptativa. Este, por sua vez, é uma extensão do algoritmo proposto por Bourgeois *et al*. (2003). O algoritmo C3_R utiliza o algoritmo GENIUS para a inserção de vértices brancos e, em seguida, utiliza uma estratégia para a inclusão dos vértices pretos à solução parcial obtida. A inclusão dos vértices pretos visa atender as restrições de cardinalidade e comprimento. O algoritmo GENIUS foi introduzido por Gendreau *et al*. (1992), onde apresenta o mesmo como a junção de dois métodos: GENI e US.

Inicialmente o procedimento GENI é aplicado considerando somente os vértices brancos. Obtida uma solução parcial incluindo todos os vértices desta cor, o algoritmo de refinamento US é aplicado na solução.

Para a inserção dos vértices pretos, seleciona-se aleatoriamente um vértice branco da solução e, em seguida, o vértice preto mais próximo a este é inserido, de forma a causar o menor incremento no custo atual da solução (à esquerda ou à direita do vértice branco). A partir deste vértice preto, a solução é percorrida verificando a posição limite que satisfaça as restrições do problema para a inserção de um outro vértice preto, o que levará à formação de uma cadeia.

Para a escolha do vértice preto é utilizado um procedimento característico da meta-heurística GRASP. É construída uma lista de candidatos (LC) contendo todos os vértices pretos não inseridos na solução e, a partir dela, é construída uma lista restrita de candidatos (LRC). Para definir os vértices que irão pertencer à LRC é utilizado o critério de qualidade dos vértices. Esse critério define o intervalo $c_{min} \leq c \leq c_{mim} + \alpha.(c_{max} - c_{min})$, no qual c equivale ao acréscimo na qualidade da solução corrente com a inserção do vértice preto. O parâmetro α controla o grau de aleatoriedade da etapa de construção e seu valor varia no intervalo [0,1]. O quanto mais próximo de zero mais guloso é o processo de seleção do vértice a ser inserido na rota, em contrapartida, quando mais próximo de 1, mais aleatória será a escolha. São incluídos na LRC os vértices cujos custos de inserção estão no intervalo. Por fim, a escolha do vértice que irá compor a rota é realizada de forma aleatória entre aqueles que estão na LRC. Este procedimento de inserção dos vértices pretos é realizado repetidas vezes até que todos os vértices estejam inseridos ou até que não seja possível formar mais cadeias que respeitam as restrições.

O C3_R termina quando não conseguir iniciar uma nova cadeia. Caso isso aconteça, se o último vértice branco foi atingido e ainda existam vértices pretos não inclusos na solução, estes são inseridos ao final da cadeia na ordem a causar o menor acréscimo possível no custo da solução. É importante ressaltar que o C3_R não garante a viabilidade da solução gerada.

3.2 Heurística de Viabilização F

A solução gerada na fase construtiva C3_R pode ser inviável, ou seja, a solução pode não satisfazer às restrições de cardinalidade ou de comprimento. Portanto, há dois tipos de inviabilidade que podem ser observadas: a inviabilidade preta e a inviabilidade branca. A primeira é causada pela violação do comprimento L de uma cadeia, enquanto na segunda há violação do número Q de vértices brancos entre dois vértices pretos consecutivos.

Na tentativa de resolver a inviabilidade preta, a estratégia é inserir um vértice preto entre dois vértices brancos da cadeia inviável. Este vértice preto será o primeiro vértice preto sucessor a esta cadeia que a torna viável. E para eliminar a inviabilidade branca a estratégia é remover os vértices brancos em excesso e inseri-los em outra cadeia que possua no máximo Q - 1 vértices brancos.

É importante ressaltar que na tentativa de eliminar a inviabilidade preta, uma inviabilidade branca pode ser gerada. Porém, ao tentar eliminar a inviabilidade branca, essa não pode violar as restrições do problema.

3.3 Buscas Locais

Na tentativa de melhorar a solução inicial, Maciel *et al*. (2005) utiliza cinco estratégias de refinamento que são aplicadas somente a soluções viáveis. O movimento da busca local é efetuado somente se não causar inviabilidade à solução.

- **BL1: Busca Local 1**

A primeira busca local (BL1) faz a realocação dos vértices. O processo é feito em cada cadeia separadamente. Dessa forma, para cada cadeia da solução, verifica-se todas as possíveis posições que cada um dos vértices brancos (inclusos na cadeia) pode assumir dentro da respectiva cadeia. Aplica-se o movimento se o custo da solução for reduzido.

- **BL2: Busca Local 2**

A BL2 é uma adaptação da fase US da heurística GENIUS, cuja diferença é que a busca local é realizada na solução contendo vértices pretos e brancos. Nesta abordagem, cada vértice da solução é removido seguindo as regras de remoção do algoritmo US e reinserido através do GENI. O movimento é aplicado somente se a nova rota mantiver a viabilidade e se houver redução no custo da solução.

- **BL3: Busca Local 3**

A BL3 trabalha com a permutação de dois vértices, indistintamente se o vértice é branco ou preto. Esse processo verifica, para cada vértice da solução, todas as possíveis trocas de posição entre este e os outros vértices. A troca será efetuada somente se houver uma melhora da solução, checando sempre a sua viabilidade.

- **BL_2-opt**

A busca local BL_2-opt é baseada na estratégia 2-opt (Cordenonsi, 2008) e foi implementada como descrita em Maciel *et al*. (2005).

- **BL4: Busca Local 4**

A BL4 é baseada na busca local utilizada por Bourgeois *et al*. 2003. A diferença é que, na busca local de Maciel *et al*. (2005), quando se obtém uma redução no custo da solução, esta assume o papel da solução corrente e a BL4 é reiniciada.

4. Heurística ILS_PCV-PB

Este trabalho propõe uma heurística baseada na metaheurística *Iterated Local Search* – ILS (Lourenço *et al*., 2010). A ILS pode ser caracterizada como uma heurística simples, robusta, eficiente e de fácil implementação (Gendreau e Jean-Yves, 2010). Devido às suas características, o método vem sendo utilizado para tratar uma variedade de problemas de otimização combinatória. Para a implementação de uma heurística ILS é necessário a definição de quatro procedimentos (Talbi, 2009; Ribeiro *et al*., 2012): (a) Construção da solução inicial, que deverá ser viável, (b) Busca Local, (c) Procedimento de Perturbação, responsável por modificar a solução corrente e, por último, (d) Critério de Aceitação, que decide qual solução será considerada a solução corrente na próxima iteração do ILS.

No Algoritmo 1 é possível observar o pseudocódigo da heurística ILS proposta para tratar o PCV-PB, cujos parâmetros de entrada são: número máximo de tentativas de construção da solução inicial viável (*nTentativas*), número máximo de iterações sem melhora (*itSM*), critério de parada do ILS, e o valor que define o grau de aleatoriedade da estratégia de construção (α).

O primeiro passo do algoritmo envolve a obtenção de uma solução que, neste algoritmo, é obtida utilizando-se a heurística C3_R (linha 1). Como mencionado anteriormente, não é possível garantir a viabilidade da solução gerada. Por isso, foi definido um número máximo de tentativas (*nTentativas*) para se obter uma solução viável (linha 3-6). Caso o algoritmo não consiga gerar uma solução viável, o algoritmo é interrompido.

Considerando que uma solução viável s_o foi gerada, nas linhas de 8-21 são executadas as demais etapas do ILS_PCV-PB. Na linha 8 uma heurística VND para o PCV-PB, estratégia que combina as buscas locais BL1, BL3, BL_2-opt e BL4, propostas por Maciel *et al*. (2005), é executada.

Em cada iteração do ILS_PCV-PB (linhas de 11 a 20), inicialmente é chamada a função *pertub_3_opt* (linha 12). A perturbação proposta é baseada na busca local 3-opt (Cordenonsi, 2008). Neste procedimento, três arestas não consecutivas são selecionadas aleatoriamente e removidas da solução atual. Em seguida, verifica-se o custo da reconexão dos fragmentos de todas as maneiras possíveis. A solução é reconstruída da forma que gerar uma solução com menor custo. Após o movimento de perturbação, o VND é chamado novamente (linha 13).

O critério de aceitação verifica o custo da solução. Como o PCV-PB é um problema de minimização, se a solução corrente *s”* tiver custo menor do que o custo da melhor solução armazenada em *s**, esta será então atualizada. Caso haja atualização, o número de iteração sem melhora do ILS é reiniciado e, na próxima iteração, a perturbação é aplicada a esta nova solução. Caso contrário, o número de iteração é incrementado. O ILS_PCV-PB termina quando o algoritmo realizar *itSM* iterações sem atualizar a solução *s**.

```
procedimento ILS_PCV-PB (nTentativas, itSM, α)
1    s_o ← C3_R(α);
2    tentativa ← 1;
3    enquanto (tentativa ≤ nTentativas e viável(s_o) = false) faça
4       s_o ← C3_R(α);
5       tentativa ← tentativa + 1;
6    fim enquanto
7    se (viável(s_o) = true) faça
8       s ← VND(s_o);
9       s* ← s;
10      iteração ← 1;
11      enquanto (iteração ≤ itSM) faça
12         s’ ← pertub_3_opt(s*);
13         s” ← VND(s’);
14         se (custo(s”) < custo(s*)) então
15            s* ← s”;
16            iteração ← 1;
17         senão
18            iteração ← iteração + 1;
19         fim se
20      fim enquanto
21      retorne s*;
22   se não
23      retorne -1;
24   fim se
fim ILS_PCV-PB;
```

Algoritmo 1: Heurística ILS proposta

5. Resultados Computacionais

Para avaliar a eficiência da heurística proposta, foi necessário gerar instâncias de diferentes tipos e dimensões para o problema abordado, uma vez que as instâncias para o PCV-PB utilizadas na literatura não foram localizadas.

Para gerar instâncias para o PCV-PB foram utilizadas como base as instâncias do Problema do Caixeiro Viajante com Agrupamentos, disponibilizadas no site do grupo Labic da UFF (http://labic.ic.uff.br/Instance/index.php).

As instâncias foram geradas de acordo com o procedimento detalhado por Maciel *et al.* (2005). Foram criadas instâncias onde a quantidade de vértices pretos foi definida em 20% ou 30% do total de vértices. O valor da cardinalidade limite Q foi calculado a partir da equação $Q = |B| / |P| + c$ em que $|B|$ é a quantidade de vértices brancos, $|P|$ a quantidade de pretos e c uma constante que pode receber valor 5 ou 10, como sugerido em Maciel *et al.* (2005). O comprimento máximo de uma cadeia é calculado por $L = z.\gamma \, / \, |P|$ onde z é o custo da solução encontrada a partir da execução do algoritmo GENIUS considerando $Q = L = \infty$ e γ assumindo valores no intervalo [1.75; 4]. A Tabela 1 mostra as características das instâncias obtidas.

Na primeira coluna da Tabela 1 tem-se a identificação das instâncias, a segunda apresenta o nome da mesma, formado pela seguinte estrutura [*nome original da instância*]_*Q* [*valor de Q*]-*L*[*valor de L*].*tsp*. Em seguida, as demais colunas apresentam, nesta ordem, as seguintes informações: a quantidade de vértices, a quantidade de vértices pretos, a quantidade de vértices brancos, o valor de γ, z, Q e L respectivamente. Ao todo foram geradas 30 instâncias contendo a maior destas 262 vértices e a menor 51 vértices.

Tabela 1: Características das instâncias.

ID	Nome da Instância	n	$\|P\|$	$\|B\|$	γ	z	Q	L
1	i-5-eil51_Q9-L114.tsp	51	10	41	2.00	574	9	114
2	i-5-eil51_Q14-L200.tsp	51	10	41	3.50	574	14	200
3	i-5-eil51_Q9-L100.tsp	51	10	41	1.75	574	9	100
4	i-5-eil51_Q12-L153.tsp	51	15	36	4.00	574	12	153
5	i-5-eil51_Q7-L114.tsp	51	15	36	3.00	574	7	114
6	i-5-eil76_Q9-L130.tsp	76	15	61	2.00	977	9	130
7	i-5-eil76_Q14-L227.tsp	76	15	61	3.50	977	14	227
8	i-5-eil76_Q9-L113.tsp	76	15	61	1.75	977	9	113
9	i-5-eil76_Q12-L177.tsp	76	22	54	4.00	977	12	177
10	i-5-eil76_Q7-L133.tsp	76	22	54	3.00	977	7	133
11	i-10-kroB100_Q14-L5168.tsp	100	20	80	2.00	51680	14	5168
12	i-10-kroB100_Q9-L9044.tsp	100	20	80	3.50	51680	9	9044
13	i-10-kroB100_Q14-L4522.tsp	100	20	80	1.75	51680	14	4522
14	i-10-kroB100_Q12-L6890.tsp	100	30	70	4.00	51680	12	6890
15	i-10-kroB100_Q7-L5168.tsp	100	30	70	3.00	51680	7	5168
16	instanciaigual-200-4-a_Q9-L1479.tsp	200	40	160	2.00	29584	9	1479
17	instanciaigual-200-4-a_Q14-L2588.tsp	200	40	160	3.50	29584	14	2588
18	instanciaigual-200-4-a_Q14-L1294.tsp	200	40	160	1.75	29584	14	1294
19	instanciaigual-200-4-a_Q7-L1972.tsp	200	60	140	4.00	29584	7	1972
20	instanciaigual-200-4_Q12-L1479.tsp	200	60	140	3.00	29584	12	1479
21	instanciaigual-200-4-x1_Q14-L1475.tsp	200	40	160	2.00	29504	14	1475
22	instanciaigual-200-4-x1_Q9-L2581.tsp	200	40	160	3.50	29504	9	2581
23	instanciaigual-200-4-x1_Q14-L1290.tsp	200	40	160	1.75	29504	14	1290
24	instanciaigual-200-4-x1_Q7-L1966.tsp	200	60	140	4.00	29504	7	1966
25	instanciaigual-200-4-x1_Q12-L1475.tsp	200	60	140	3.00	29504	12	1475
26	i-9-gil262.tsp-3x3_Q14-L139.tsp	262	52	210	2.00	3629	14	139
27	i-9-gil262.tsp-3x3_Q9-L244.tsp	262	52	210	3.50	3629	9	244
28	i-9-gil262.tsp-3x3_Q14-L122.tsp	262	52	210	1.75	3629	14	122
29	i-9-gil262.tsp-3x3_Q7-L186.tsp	262	78	184	4.00	3629	7	186
30	i-9-gil262.tsp-3x3_Q12-L139.tsp	262	78	184	3.00	3629	12	139

Para definir o valor de α foi realizada uma calibração empírica onde foram testados dez valores {0.1, 0.2, ..., 0.9, 1}. Para cada valor testado, cada uma das instâncias foi executada 10 vezes sendo observada a média da qualidade das soluções obtidas nessas execuções. Diante dos resultados, foi selecionado α = 0.4. Os demais parâmetros foram ajustados com número máximo de iterações sem melhora igual a 100 e o número máximo de tentativas de construção da solução inicial definido como 20.

Com o objetivo de verificar a eficiência da heurística proposta, foi implementada também a heurística GRASP-VND de Maciel *et al*. (2005) que apresentou os melhores resultados entre as heurísticas da literatura.

As implementações foram feitas na linguagem C++ e os testes computacionais foram realizados em uma máquina com processador Intel Core i5 de 3.40 GHz e RAM de 16 GB.

Em testes iniciais, verificou-se que o tempo de processamento das heurísticas estava sendo dominado pelo tempo consumido pela estrutura de vizinhança BL2. Para mostrar o impacto do uso desta busca local, na Tabela 2 pode-se observar os resultados do ILS_PCV-PB considerando ou não a BL2. Para esta avaliação foram selecionadas 10 instâncias, sendo que, para cada uma destas, o ILS_PCV-PB foi executado 10 vezes. A primeira coluna da Tabela 2 apresenta o identificador das instâncias. Na segunda coluna observa-se a qualidade da melhor solução considerando todas as execuções destes dois algoritmos. Na terceira e quarta colunas são apresentados os resultados para o ILS_PCV-PB considerando o uso da BL2. Foi calculada a média das 10 soluções encontradas por esta abordagem para cada uma das instâncias e, depois disso, verificou-se a diferença percentual desta média em relação à melhor solução (terceira coluna). O tempo médio de processamento, em segundos, é apresentado na quarta coluna. Na quinta e na sexta coluna são apresentados estes mesmos valores para a abordagem ILS_PCV-PB que não faz uso da BL2.

Como pode-se observar na Tabela 2, usando a BL2 a qualidade das soluções obtidas pela ILS_PCV-PB é muito próxima da melhor solução obtida, ficando a apenas 0,04% deste valor. Apesar disso, com o uso da BL2 o tempo de processamento é mais de 100 vezes maior. Enquanto a estratégia que não utiliza a BL2 consome em média 0,5 segundos, com o uso da BL2 este tempo sobe para 57,21 segundos. Como o ganho na qualidade não foi tão significativo, já que sem utilizar a BL2 os resultados ficam em média a 0,56% da melhor solução, nos demais testes apresentados neste trabalho não é utilizada a BL2.

Tabela 2 - Resultados computacionais do ILS_PCV-PB com a BL2 e sem a BL2.

ID	Melhor solução	ILS com BL2		ILS sem BL2	
		dif %	tempo (s)	dif %	tempo (s)
1	435,40	0,03%	27,56	0,56%	0,23
2	428,87	0,09%	23,41	0,41%	0,27
3	436,87	0,00%	23,69	0,45%	0,24
4	428,87	0,02%	17,52	0,64%	0,19
5	428,87	0,02%	19,39	0,37%	0,24
6	553,63	0,01%	104,06	0,74%	0,72
7	545,97	0,20%	104,93	0,63%	0,72
8	553,63	0,01%	95,79	0,51%	1,03
9	544,37	0,00%	67,02	0,74%	0,68
10	549,57	0,06%	88,72	0,59%	0,71
Média		0,04%	57,21	0,56%	0,50

Para uma comparação entre o ILS_PCV-PB proposto e o algoritmo de Maciel *et al*. (2005), cada um dos métodos foi executado 10 vezes para cada uma das 30 instância e os resultados são apresentados na Tabela 3. Para este experimento, todos os parâmetros do algoritmo da literatura, GRASP_VND, foram ajustados como sugerido em Maciel *et al*. (2005).

Na Tabela 3 observa-se na coluna ID o identificador da instância e na segunda coluna, rotulada *Melhor solução*, a qualidade da melhor entre as soluções observadas considerando-se todas as execuções dos algoritmos avaliados. Para a estratégia proposta neste trabalho ILS_PCV-PB e para a heurística da literatura GRASP_VND, na terceira e na quinta colunas são apresentadas as diferenças percentuais entre a média da qualidade observada para cada instância e o valor da melhor solução. Além desta informação, os tempos médios de execução são apresentados na quarta e sexta colunas, respectivamente, para o ILS_PCV-PB e GRASP_VND.

Tabela 3 - Comparação entre ILS_PCV-PB e GRASP_VND

ID	Melhor solução	ILS_PCV-PB		GRASP_VND	
		dif %	tempo (s)	dif %	tempo (s)
1	435,40	0,56%	0,23	1,37%	6,54
2	428,87	0,41%	0,27	2,18%	6,42
3	436,87	0,45%	0,24	1,28%	6,33
4	428,87	0,64%	0,19	1,85%	4,38
5	428,87	0,37%	0,24	1,31%	4,45
6	552,46	0,95%	0,72	3,00%	23,62
7	545,97	0,63%	0,72	2,07%	23,61
8	553,03	0,62%	1,03	3,29%	23,39
9	544,37	0,74%	0,68	3,06%	15,81
10	549,57	0,59%	0,71	3,32%	15,37
11	22139,10	0,36%	1,57	1,81%	54,04
12	22593,90	0,97%	2,11	4,20%	53,64
13	22139,10	0,32%	1,59	1,16%	53,20
14	22507,20	1,61%	1,37	2,57%	40,05
15	22507,20	0,43%	1,64	2,58%	39,01
16	11108,50	1,47%	29,55	5,88%	1346,17
17	10950,40	1,59%	22,10	4,65%	1333,99
18	10950,40	1,34%	24,24	4,54%	1339,29
19	11053,00	1,59%	18,72	5,17%	796,39
20	10935,30	2,07%	14,35	4,63%	798,26
21	10950,40	1,57%	17,86	4,86%	1333,59
22	11108,50	1,23%	33,50	5,42%	1378,71
23	10950,40	1,37%	22,64	4,74%	1330,51
24	11053,40	1,26%	17,46	5,07%	812,26
25	10935,30	1,71%	16,37	4,88%	827,16
26	1222,43	1,48%	50,01	4,19%	2821,17
27	1228,30	2,82%	57,47	5,71%	2810,77
28	1221,57	1,53%	45,94	4,31%	2827,94
29	1229,15	1,61%	46,33	5,77%	1799,61
30	1220,52	1,12%	41,80	5,04%	1812,44
Média		1,11%	15,72	3,66%	791,27

Como é possível observar na última linha da Tabela 3, o ILS proposto apresenta resultados que ficam em média a 1,11% distantes da melhor solução conhecida. Já a heurística GRASP_VND fica em média a 3,66% deste valor. Para todas as instâncias a solução média obtida pelo ILS é melhor que a solução obtida pela estratégia da literatura. Em relação ao tempo de processamento, é possível verificar que o ILS é cerca de 50 vezes mais rápido do que a estratégia da literatura.

6. Conclusões

Este trabalho aborda uma generalização do clássico PCV, chamado de Problema do Caixeiro Viajante Preto e Branco, PCV-PB, pertencente a classe de problemas NP-difíceis. No PCV-PB cada vértice é classificado como Preto ou Branco. Além das restrições do PCV, duas restrições adicionais são consideradas, uma relacionada ao número máximo de vértices brancos entre dois vértices pretos e uma outra que determina uma distância limite entre dois vértices pretos. O PCV-PB possui aplicações, por exemplo, no campo de agendamento de companhias aéreas e na área de telecomunicações.

Para tratar o problema foi proposta uma heurística ILS, chamada ILS_PCV-PB, que combina diversas estratégias de busca local descritas na literatura. Para avaliar a qualidade da estratégia proposta, os resultados obtidos foram comparados com os resultados da heurísticas GRASP_VND da literatura, proposta por Maciel *et al*. (2005).

Para os testes computacionais foram geradas 30 instâncias onde o número de vértices variou entre 51 e 256. Para a geração das instâncias foi observado o procedimento proposto em (Maciel *et al*., 2005). Estas instâncias foram obtidas a partir de instâncias do Problema do Caixeiro Viajante com Agrupamentos.

Na comparação entre os resultados obtidos pela heurística ILS_PCV-PB proposta e pela heurística GRASP-VND da literatura, pode-se observar que a heurística ILS obtém soluções de melhor qualidade e destaca-se pela redução no tempo de processamento necessário para sua execução

Como trabalhos futuros, considera-se a possibilidade de desenvolvimento de novas estratégias para obtenção de uma solução inicial e também a utilização de outras heurísticas de refinamento. Após este aprimoramento das heurísticas básicas, outras metaheurísticas podem ser avaliadas.

Agradecimentos

Os autores agradecem a FAPEMIG pelo financiamento parcial deste trabalho.

Referências

Bhattacharya, B., **Hu**, Y e **Kononov**, A. (2007) Approximation Algorithms for the Black and White Traveling Salesmam Problem. Anais do *Computing and Combinatorics, 13th Annual International Conference*, Springer: Berlin Heidelberg, p. 559-567.

Bourgeois, M., Laporte, G. e Semet, F. (2003), Heuristics for the black and white traveling salesman problem. *Computers & Operations Research*, 30, 1, p. 75-85.

Cordenonsi, A. Z. Ambientes. Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em Heurísticas e Metaheurísticas. Universidade Federal do Rio Grande do Sul. *Tese de doutorado em Informática na Educação*. 2008 (http://www.lume.ufrgs.br/).

Cosares, S., Deutsch, D. N., Saniee, I. e Wasem, O. J. (1995), SONET toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*. 25. 1 p. 20-40.

Garey, R. M. e Johnson, S. D. Computers and Intractability: A guide to the theory of NP-completeness. W. H. Freeman & Company. San Francisco, 1979.

Gendreau, M., Hertz, A. e Laporte. G. (1992), New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40, 6, p. 1086-1094.

Ghiani, G., Laporte, G. e Semet, F. (2006), The black and white traveling salesman problem. *Operations Research*, 54, 2, p. 366-378.

Jiang, H., **Zhang**, X., **Li**, M. e **Che**, H. Y. (2007) Using Gavish-Grave LP to Formulate the Directed Black and White Traveling Salesman Problem. Anais do *7° International Conference On Computational Science (ICCS)*, Springer: Berlin Heidelberg, p. 293-298.

Lopes, H. S., Rodrigues, L. C. A. e Steiner, M. T. A. *Meta-Heurísticas em Pesquisa Operacional*. Omnipax, Curitiba, 2013.

Lourenço, H. R., Martin, O. C. e Stutzle, T. Iterated local search: Framework and Applications. *In Handbook of metaheuristics*. Springer. New York. 146. p. 363-397, 2010.

Maciel, A. C. M., Martinhon, C. A. e Ochi, L. S. (2005), Heurísticas para o Problema do Caixeiro Viajante Branco e Preto. Anais do *XXXVII SBPO,* 1349-1360.

Mak, V. e Boland, N. (2000), Heuristic approaches to the asymmetric travelling salesman problem with replenishment arcs. *International Transactions in Operational Research*, 7, 4-5, p. 431–447.

Ramos, I. C. O., Goldbarg, M. C. e Goldbarg, E. F. G. (2003), A ProtoG algorithm applied to the traveling salesman problem. Anais do *XXIII International Conference of the Chilean Computer Society*, p. 23-30.

Ribeiro, W. G., Arroyo, J. E. C. e Santos, A. G. (2012), Metaheurística ILS para O Problema De Cobertura e Roteamento em Redes de Sensores Sem Fio. *Anais do XVI Congresso Latino-Iberoamericano de Investigación Operativa,* p. 2676-2587.

Talbi, El-G. *Metaheuristics: From Design to Implementation*. John Wiley & Sons. New Jersey, 624, p. 593, 2009.

Talluri, K. T. (1998). The four-day aircraft maintenance routing problem. *Transportation Science*, 32, 1, p. 43–53.

Wasem, O. J. (1991), An algorithm for designing rings in survivable fiber networks. *IEEE Trans. Reliability*, 40, 4, p. 428–432.