

# Resolução do Problema do Caixeiro Viajante e Tomada de Decisão Multicritério

Ariel François Maia Domingues  
Engenharia de Sistemas

Universidade Federal de Minas Gerais  
Belo Horizonte, MG 31255-320  
Email: ariel.f.m.domingues@gmail.com

Hernane Braga Pereira  
Engenharia de Sistemas

Universidade Federal de Minas Gerais  
Belo Horizonte, MG 31270-901  
Email: hernanepereira@hotmail.com

Nikolas Dias Magalhães Fantoni  
Engenharia de Sistemas

Universidade Federal de Minas Gerais  
Belo Horizonte, MG 31270-901  
Email: nikolasfantoni@gmail.com

**Resumo**—Este artigo<sup>1</sup> tem por finalidade apresentar uma solução para o Problema do Caixeiro Viajante a partir da metaheurística *Simulated Annealing Multi-Objetivo*, visando otimizar a distância percorrida e o tempo do trajeto do Caixeiro, e realizar uma tomada de decisão auxiliada, utilizando os métodos AHP, ELECTRE I e PROMETHEE II, para escolher uma solução dentre as apresentadas na etapa de otimização. A linguagem utilizada para a implementação dos algoritmos foi o R e os resultados mostraram uma boa otimização multi-objetivo dos algoritmos em relação a uma solução inicial e, a partir das soluções otimizadas encontradas, foi possível decidir uma solução preferida dentre o grupo apresentado, mostrando a utilidade dos métodos de auxílio à tomada de decisão.

**Palavras-Chave**—Problema do Caixeiro Viajante, Otimização Multi-objetivo, Tomada de Decisão Multicritério.

## I. INTRODUÇÃO

O problema do caixeiro viajante (PCV) é um problema clássico na literatura, onde um entregador ou vendedor deve percorrer o caminho de menor custo a partir de uma cidade inicial, visitando uma única vez todas as outras  $n$  cidades, e então retornar para o ponto de partida. A solução do problema é combinatória e ele é classificado como NP-completo, pois não há como resolver esse problema utilizando um algoritmo de ordem de complexidade polinomial. Assim, são aplicadas meta-heurísticas, de forma a tentar otimizar a solução do problema para um valor considerado ótimo, dadas determinadas métricas inicialmente definidas. [1]

Para este trabalho será usada uma instância de 250 cidades, todas interligadas entre si (*i.e.* pode-se ir, a partir de qualquer cidade A, para qualquer cidade B, com  $A \neq B$ ), de valores assimétricos, ou seja, apesar de a distância da cidade A até B ser idêntica a de B até A, o tempo entre os deslocamentos de ida e de volta podem ser diferentes. Nesta etapa do trabalho, foram fornecidos dois arquivos \*.csv contendo os valores das distâncias e tempos entre todas as cidades do problema. Cabe ressaltar que a distância e o tempo entre uma cidade e ela mesma é  $d = t = 0$ .

O problema será tratado como multi-objetivo e, portanto, a função objetivo será uma minimização simultânea da distância

total e do tempo total gastos pelo Caixeiro, através da técnica de meta-heurística *Simulated Annealing* (SA) e utilizando um total de seis estruturas de vizinhança alternadas, a fim de obter um resultado mais próximo de um possível ótimo global e evitar resultados ótimos locais. A abordagem utilizada na otimização multi-objetivo foi a da Soma Ponderada ( $P_w$ ) para definição das alternativas finais.

Por fim, será escolhida uma rota dentre as alternativas, a partir de uma tomada de decisão auxiliada, que será a solução final para o problema. Para a tomada de decisão serão implementado o método *Analytic Hierarchy Process* (AHP), para a definição dos pesos dos critérios a serem utilizados na tomada de decisão, além dos métodos ELECTRE I e PROMETHEE II, para auxiliar na tomada de decisão final.

## II. METODOLOGIA

Todos os algoritmos aqui apresentados foram implementados na linguagem R, um ambiente de desenvolvimento para aplicações estatísticas e gráficas<sup>2</sup>. Para entender a representação do problema e a solução proposta, são necessários o entendimento de alguns tópicos, explicados posteriormente nesta sessão: a descrição matemática do problema do PCV, descrita em *Modelagem do PCV*; a descrição do algoritmo de otimização utilizado, descrito em *Simulated Annealing* (SA), a descrição do algoritmo de solução inicial proposto, descrito em *Solução Inicial*, a descrição das estruturas de vizinhanças propostas para a resolução do problema, encontradas em *Estruturas de Vizinhanças*, uma introdução à teoria da decisão auxiliada, descrita em *Tomada de Decisão* e por fim a explicação dos métodos de tomada de decisão implementados, descritos nas sessões AHP, ELECTRE I e PROMETHEE II.

### A. Modelagem do PCV

O problema do caixeiro viajante, também conhecido como *Traveling-Salesman Problem* (TSP), é um dos problemas que mais recebe atenção para sua resolução, visto que há grande aplicabilidade na indústria, devido ao seu grande potencial de retorno econômico. O PCV foi inicialmente abordado por Dantzig's [1] como um problema de otimização linear e solucionado utilizando o método simplex para uma instância

<sup>1</sup>Trabalho realizado para a disciplina ELE088 - Teoria da Decisão, ministrada pelo professor Lucas de Souza Batista do Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais para o segundo semestre do ano de 2019.

<sup>2</sup>Mais sobre o R em: <https://www.r-project.org/about.html>

de 49 cidades. Entretanto, por se tratar de um problema combinatório, esta abordagem não é viável ao ser utilizada em larga escala, portanto atualmente são mais comuns abordagens do problema utilizando de meta-heurísticas, que apesar de não garantirem otimalidade, entregam bons resultados em um tempo computacional viável. O problema é representado como um grafo, onde cada vértice representa uma cidade e cada aresta é equivalente a uma viagem da cidade  $i$  até  $j$ . Aqui, o grafo será representado como uma estrutura de matriz de adjacências, com o tamanho  $[n \times n]$ . Esta estrutura foi escolhida, pois todas as cidades estão interligadas entre si. A figura 1 mostra um exemplo de um grafo.

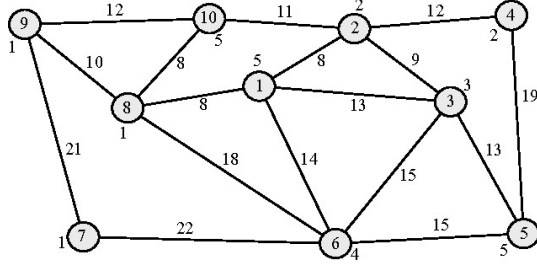


Fig. 1. Exemplo de um grafo.

Uma das maneiras de modelar o problema [2] encontra-se a seguir:

$$\min \sum_{j=1}^m \sum_{i=1}^m c_{ij} x_{ij} \quad (1)$$

$$\text{Sujeito a : } \sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j = 1, \dots, m \quad (3)$$

$$\sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1 \quad \forall i, j \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \quad (5)$$

onde  $K$ , é um subconjunto não vazio das cidades  $1, \dots, m$ . Para o caso onde será avaliado o tempo entre cidades, adiciona-se a restrição:

$$d_{ij} \neq d_{ji} \quad \forall i, j \quad (6)$$

Pois apesar das distâncias entre cidades serem iguais, o tempo entre elas é diferente, variando de acordo com o ponto de partida.

## B. Simulated Annealing (SA)

Para a resolução do PCV será o utilizado o método do *Simulated Annealing* (SA) multi-objetivo baseado em vizinhanças. O algoritmo SA pertence à classe dos algoritmos de busca local e é muito importante por possuir uma componente estocástica envolvida, o que facilita sua análise de convergência, além de ser aplicável a uma vasta quantidade de problemas. Sua maior vantagem é que a busca é feita procurando sempre aproximar do ótimo global, ao invés de aceitar soluções ótimas locais, o que é possível pois o algoritmo aceita soluções intermediárias cujo custo da função objetivo pode ser maior que o de uma solução intermediária anteriormente encontrada, para então refinar a pesquisa em busca de um ótimo. O algoritmo baseia-se na ideia do resfriamento lento de peças metálicas, usado na indústria metalúrgica com o objetivo de obter características como dureza e tenacidade do material diferentes das obtidas se fosse utilizado o resfriamento rápido [3] [4].

Para entender a implementação geral do algoritmo, são necessárias algumas definições abaixo:

- Temperatura inicial  $T_{0d}$  e  $T_{0t}$ : define o grau de aceitação de soluções cujo resultado da função objetivo é pior que a solução anterior, para a distância e para o tempo, de forma separada.
- Iterações por temperatura  $m$ : define a quantidade de iterações que serão feitas a cada temperatura definida.
- Função de mudança de temperatura  $f(T_k)$ : define a temperatura  $T_{k+1}$  a partir de um  $T_k$  dado.
- Iterações totais  $n$ : define o total de mudanças de temperatura máximas possíveis.
- Função de Vizinhanças  $\aleph(\bar{x}, lvl)$ : gera uma solução  $\bar{x}_2$ , com nível de perturbação  $lvl$ , a partir de uma solução  $\bar{x}$ .
- Função de solução inicial  $init()$ : gera uma solução inicial para o problema.
- Nível da vizinhança  $lvl$ : define o nível de perturbação possível que cada vizinhança pode fazer, sendo o nível 1 considerado perturbações leves (que geram soluções parecidas com a da entrada) até o nível 6 (que gera perturbações consideravelmente distintas da solução de entrada).
- Diferença dos custos  $\Delta E$ : define a diferença entre os custos da solução atual em relação ao tempo e à distância, simultaneamente, e da solução calculada a cada iteração.
- Função aleatória  $rnd(0..1)$ : gera um número aleatório entre 0 e 1 para comparar à função de probabilidade de aceitação para uma solução cujo custo é menor que o custo da solução atual.

O algoritmo do SA utilizado é uma variação do algoritmo clássico mostrado na literatura, mostrado a seguir. A variação consiste no uso de seis estruturas de vizinhança, classificadas pelo nível de perturbação  $lvl$ , que são escolhidas de acordo com a quantidade de soluções aceitas por iteração. Ou seja, se o algoritmo estiver com dificuldades de encontrar uma nova solução utilizando a estrutura de nível baixo, ele aumenta o nível para tentar encontrar uma nova solução. Assim, a melhor

descrição do algoritmo implementado neste trabalho consiste em:

- 1) procura uma solução inicial e a considera a melhor;
- 2) encontra uma solução na vizinhança, com um nível de perturbação baixo;
- 3) calcula o  $\Delta E$ ;
- 4) se  $\Delta E$  for menor ou igual a zero, a solução da vizinhança se torna a melhor até então e substitui a solução analisada. Se  $\Delta E$  for maior que zero, um número aleatório é gerado, entre 0 e 1. Se for menor que a função de probabilidade  $p = e^{\frac{-\Delta E_d}{T_{kd}}} \cdot e^{\frac{-\Delta E_t}{T_{kt}}}$ , então a solução é aceita e substituída como a solução a ser analisada, caso contrário nada acontece;
- 5) repete os itens de 2 ao 4 para o total de  $m$  iterações;
- 6) atualiza a temperatura e o nível, este caso nenhuma solução tenha sido aproveitada utilizando o nível anterior;
- 7) repete os itens do 2 ao 6 até o total de  $n$  iterações ou até a temperatura ser suficientemente baixa;
- 8) retorna a melhor solução encontrada.

A função de variação de temperatura é dinâmica [5] [6], mostrada na equação 7, em que  $\bar{E}$  corresponde a média de todas as diferenças de custos entre as soluções aceitas e as soluções atuais,  $\min(E)$  corresponde à menor diferença encontrada e  $D_0$  é um valor fixado (entre 0,5 e 0,9). Assim, decide-se o menor valor entre a razão entre o custo mínimo e o custo médio ou entre o valor fixo  $D_0$  e multiplica-se pela temperatura atual para obter a próxima temperatura. Esta operação é feita para cada temperatura, relacionada à distância e ao tempo, separadamente.

$$T_{k+1} \leftarrow \min\left(\frac{\min(E)}{\Delta \bar{E}}, D_0\right) \times T_k \quad (7)$$

A temperatura inicial foi calculada conforme a equação 8, em que o coeficiente  $\tau_0$  é uma taxa de aceitação inicial. Para este problema, foi considerado  $\tau_0 = 0,5$ . Assim, são feitas 100 pequenas perturbações na solução inicial, obtendo-se uma série de  $\Delta E$ . Após isso, obtém-se a média  $\Delta \bar{E}$ . Foram calculadas duas temperaturas iniciais distintas, um  $T_{0d}$  para as distâncias e um  $T_{0t}$  para o tempo.

$$e^{\frac{-\Delta \bar{E}}{T_0}} = \tau_0 \quad (8)$$

Foi necessária ainda uma normalização da função de custo, uma vez que as dimensões da distância destoam muito em relação ao tempo, o que pode prejudicar no balanceamento da otimização. A função normalizada é mostrada na equação 9, uma vez que o problema é de minimização.

$$f_i(x) = \frac{f_i(x) - \min f_i(x)}{\max f_i(x) - \min f_i(x)} \quad (9)$$

Para a abordagem de soma ponderada, a função objetivo passa a ser um somatório ponderado de cada função objetivo, isto é

$$\min w_t f_t(\bar{x}) + w_d f_d(\bar{x}), \quad (10)$$

---

### Algoritmo 1: SIMULATED ANNEALING

---

**Entrada:**  $f(T_k)$ ,  $T_{0d}$ ,  $T_{0t}$ ,  $init()$ ,  $\aleph(\bar{x}, lvl)$

**Saída:** Melhor solução  $x^*$

```

1  início
2   $\bar{x}_1 \leftarrow init()$ 
3   $\bar{x}^* \leftarrow \bar{x}_1$ 
4   $T_{kd} \leftarrow T_{0d}$ 
5   $T_{kt} \leftarrow T_{0t}$ 
6   $lvl \leftarrow 1$ 
7  para cada  $i \in n$  faça
8  |    $aceito \leftarrow 0$ 
9  |   para cada  $j \in m$  faça
10 | |    $x_2 \leftarrow \aleph(\bar{x}_1, lvl)$ 
11 | |    $\Delta E \leftarrow \text{custo norm. } \bar{x}_2 - \text{custo norm. } \bar{x}_1$ 
12 | |   se  $\Delta E \leq 0$  então
13 | | |    $aceito \leftarrow aceito + 1$ 
14 | | |    $\bar{x}^* \leftarrow \bar{x}_2$ 
15 | | |    $\bar{x}_1 \leftarrow \bar{x}_2$ 
16 | |   fim
17 |   senão
18 | |   se  $rnd(0..1) < e^{\frac{-\Delta E_d}{T_{kd}}} \cdot e^{\frac{-\Delta E_t}{T_{kt}}}$  então
19 | | |    $\bar{x}_1 \leftarrow \bar{x}_2$ 
20 | | |    $aceito \leftarrow aceito + 1$ 
21 | |   fim
22 |   fim
23 fim
24  $T_k \leftarrow f(T_k)$ 
25 se  $aceito = 0$  então
26 |    $lvl \leftarrow lvl + 1$ 
27 fim
28 senão
29 |    $lvl \leftarrow 1$ 
30 fim
31 fim
32 fim
33 retorna  $\bar{x}^*$ 

```

---

em que  $w_i \forall i \in \{d, t\}$  é o peso considerado para a otimização da distância e do tempo, respectivamente, e  $f_i \forall i \in \{d, t\}$  são as funções dos custos da distância e do tempo, também respectivamente. Serão utilizados  $w_d = w_t = 0,5$ , uma vez que não há preferência *a priori* de otimização entre o tempo e a distância. Assim, ambas as restrições são otimizadas simultaneamente e o algoritmo normaliza a função de custo para definir se possui uma solução  $x^*$  melhor que a anterior.

Para a tomada de decisão, foram primeiro necessárias as definições das soluções Pareto-Ótimas do problema, que são soluções que não dominam outras soluções pertencentes a este conjunto. A dominância ocorre se, para uma solução  $x$  do conjunto de alternativas, existe outra solução  $y$  cujo valor das funções objetivos são no mínimo equivalentes ao valor da solução  $x$  e em ao menos um objetivo a solução  $x$  é melhor que a solução  $y$ . Desta forma, dizemos que  $x$  domina  $y$  [7].

O conjunto de soluções Pareto-Ótimas foi obtido variando

os pesos da abordagem da soma ponderada, gerando várias soluções para o problema que posteriormente foram comparadas entre si, a fim de eliminar soluções dominadas. Ao final restaram X soluções que compõem nosso conjunto eficiente, que será usado na etapa de tomada de decisão.

### C. Solução Inicial

A função de solução inicial usada no SA retornar uma matriz de tamanho  $[3 \times 250]$  de cidades visitadas, onde a coluna  $j$  representa a cidade de origem e as linhas  $i$  representam a cidade de destino e o custo até a mesma. A função calcula a solução da seguinte forma: para cada cidade visitada, o custo entre cidades é ordenado de forma crescente e a próxima cidade é escolhida aleatoriamente entre as  $r$  cidades mais próximas, onde  $r$  é um parâmetro de entrada da função e quanto maior o valor de  $r$ , mais aleatória será a solução inicial gerada. Ao final da função, é calculada a distância para se retornar à cidade de origem. Como a função não é determinística, ela foi executada 5 vezes para verificar sua robustez com valores de  $r = 3$  e  $r = 10$ . Os resultados são mostrados na tabela I.

TABLE I  
RESULTADOS DOS TESTES PARA O ALGORITMO DA SOLUÇÃO INICIAL

TESTE	r=3		r=10	
	DISTÂNCIA	TEMPO	DISTÂNCIA	TEMPO
1	2432,8	44,1	4131,7	71,8
2	2608,7	44,6	4138,9	72,8
3	2366	42,5	4427,1	69,4
4	2583,1	43,7	4338,9	72,5
5	2579,1	43,5	4168,5	69,2
Média	2513,94	43,68	4241,02	71,14
Desvio Padrão	107,79	0,78	134,02	1,72

### D. Estrutura de Vizinhanças

As estruturas de vizinhança são peças fundamentais no funcionamento esperado do algoritmo SA. Elas são as responsáveis por modificar as soluções candidatas de modo a navegar pelo espaço de alternativas em busca dos ótimos locais, com o objetivo de mapear melhores soluções e de preferência o ótimo global. [8]

É preciso haver um equilíbrio nas estruturas de vizinhança. Elas devem modificar a solução candidata o suficiente para o prosseguimento do algoritmo, porém não tanto para que ocorram saltos grandes muito frequentemente, o que atrapalha o refinamento de ótimos locais.

O ideal, portanto, é ter várias estruturas de vizinhança com níveis de perturbação diferentes, para cada ocasião. Níveis de perturbação baixos são melhores para refinar soluções em uma mesma bacia que contém um ótimo local. Porém, quando o SA passa a não melhorar a solução é preciso buscar sair dessas bacias em busca de novas soluções, potencialmente melhores. Assim, níveis de perturbação crescentes são definidos, para que seja possível fazer esses saltos entre bacias. Depois o nível pode ser diminuído novamente para refinamento da nova bacia encontrada, e o processo se reinicia.

No algoritmo proposto, foram definidos seis estruturas de vizinhança. A ordem de perturbação é definida analisando dois critérios: o número de arestas modificadas (são elas que contém o custo de ir de uma cidade a outra) e a distância para seus vizinhos (quanto mais longe o vizinho, maior é a perturbação). As estruturas, de menor nível de perturbação para maior, são: Troca Vizinha Simples, Deslocamento Simples, Inversão, Troca Vizinha Dupla, Troca Intervalada e Deslocamento Duplo.

- **Troca Vizinha Simples:** Estrutura de nível de perturbação 1. Uma cidade é escolhida aleatoriamente e troca de lugar com seu vizinho da frente. Ou seja, se a ordem de um caminho for  $A > B > C > D > E$ , e B é a cidade escolhida, então B troca com C e a nova ordem passa a ser  $A > C > B > D > E$ .

Na troca, são modificadas 3 arestas (3 retiradas e trocadas por outras 3) e o vizinho é o imediato da frente (baixa perturbação). Dessas 3 arestas, 1 só inverte seu sentido (se fosse de A para B, passa a ir de B para A), o que quer dizer que se a matriz de custos for simétrica (custo de A para B igual a custo de B para A), apenas 2 arestas afetam de fato o custo da nova solução.

- **Deslocamento Simples:** Estrutura de nível de perturbação 2. Uma cidade é escolhida aleatoriamente e sofre um deslocamento para frente de 3 a 7 cidades (probabilidade do tamanho do deslocamento distribuída uniformemente). Ou seja, se a ordem do caminho for  $A > B > C > D > E > F$ , e B é a cidade escolhida, então B é deslocada e a ordem passa a ser  $A > C > D > E > F > B$ , se o deslocamento for de 4 cidades por exemplo.

O deslocamento modifica 3 arestas, assim como o anterior, porém não é afetado pela simetria dos custos. Além disso, os vizinhos são mais distantes (em média 5 cidades de distância), aumentando a perturbação.

- **Inversão:** Estrutura de nível de perturbação 3. Uma cidade é escolhida aleatoriamente e tem o trecho subsequente de 3 a 7 cidades invertido (probabilidade do tamanho do trecho distribuída uniformemente). Ou seja, se o caminho for  $A > B > C > D > E > F$ , e B é a cidade escolhida, então o caminho seguinte a B é invertido de forma a se tornar  $A > E > D > C > B > F$ , se o trecho for de 4 cidades por exemplo.

No pior caso, são modificadas 8 arestas, porém 6 delas apenas com inversão de sentido, o que reduz a 2 arestas a perturbação para matrizes de custo simétricas. Para matrizes assimétricas, a perturbação passa a ser mais significativa, porém, como a inversão ocorre com cidades vizinhas, a perturbação é baixa, mesmo em relação a outras estruturas que alteram menos arestas que esta.

- **Troca Vizinha Dupla:** Estrutura de nível de perturbação

4. É idêntica à estrutura de nível 1, com a diferença que neste caso ocorrem duas trocas aleatórias ao invés de uma só. Assim, 6 arestas são modificadas, sendo 2 invertidas apenas, mas ambas trocas ocorrem com vizinhos imediatos.

- *Troca Intervalada*: Estrutura de nível de perturbação 5. Uma cidade é escolhida aleatoriamente e é trocada de lugar com outra cidade a sua frente, com um intervalo de 3 a 7 cidades entre elas (probabilidade do tamanho do intervalo distribuída uniformemente). Ou seja, se o caminho for  $A > B > C > D > E > F > G$ , e B é a cidade escolhida, então ocorre a troca e o novo caminho passa a ser  $A > G > C > D > E > F > B$ , se o intervalo for de 4 cidades por exemplo. Nesta troca, 4 arestas são modificadas e a distância entre os vizinhos é grande (em média 5 cidades), o que representa uma perturbação superior (em média) às anteriores.
- *Deslocamento Duplo*: Estrutura de nível de perturbação 6. É idêntica à estrutura de nível 2, com a diferença que neste caso ocorrem dois deslocamentos aleatórios ao invés de um só. Assim, 6 arestas são modificadas, e as trocas ocorrem com vizinhos distantes (de 5 cidades em média). Desta forma, é a estrutura que tende a perturbar mais as soluções candidatas.

#### E. Tomada de Decisão

Apesar de que a maioria das decisões tomadas em situações diárias são rotineiras e não necessitam de uma análise profunda das alternativas, algumas vezes encontramos casos em que se faz necessário o uso de software para auxiliar na tomada de decisão. Uma decisão errada pode acarretar em danos físicos, financeiros e sociais para organizações empresariais, pessoais ou mesmo governamentais. Para uma boa tomada de decisão, são necessários conhecer três elementos principais [9]:

- Um conjunto de alternativas  $X$ , que representam as possíveis opções para o Tomador de Decisões (DM) escolher.
- Um conjunto de critérios  $f = (f_1, f_2, \dots, f_q)$ , com os quais preocuparemos para tomar uma boa decisão.
- Um conjunto de saídas  $f(x) = (f_1(x), f_2(x), \dots, f_q(x))$ , que são baseadas nos critérios  $f$  e cujos valores são métricas para a tomada de decisão.

Definimos 5 critérios que consideramos importantes em uma boa solução. Logicamente, os critérios de menor distância total e menor tempo total foram selecionados, visto que esses são os objetivos de fato do caixeiro viajante proposto.

Escolhemos outros 3 critérios que estariam relacionados ao trabalho do vendedor/entregador que teria que percorrer as cidades. Pensamos que seria mais conveniente ter uma solução com menor variação tanto entre as distâncias quanto entre os tempos de cidade a cidade, de forma que seria mais fácil controlar o encaixe das viagens no horário de trabalho da pessoa. Assim, os dois próximos critérios são o menor desvio padrão da distância e o menor desvio padrão de tempo.

Por fim, colocamos um critério para definir um tempo limite de viagem entre cidades para que o trabalhador possa trabalhar com um mínimo de entregas em seu dia de trabalho. Escolhemos, portanto, um tempo limite de 1.3h de viagem entre as cidades, que representaria um mínimo de 3 cidades visitadas nas 8h de trabalho (considerando que a descarga e recarga demorem em torno de 1.3h por entrega).

Assim, os critérios contínuos podem ser escritos matematicamente como:

$$\begin{aligned} c_1(\mathbf{A}) &= \sum_{i=1}^{250} d_{ij} \\ c_2(\mathbf{A}) &= \sum_{i=1}^{250} t_{ij} \\ c_3(\mathbf{A}) &= \sqrt{\frac{\sum_{i=1}^{250} (d_{ij} - \bar{d})^2}{250}} \\ c_4(\mathbf{A}) &= \sqrt{\frac{\sum_{i=1}^{250} (t_{ij} - \bar{t})^2}{250}}, \end{aligned} \quad (11)$$

em que  $\mathbf{A}$  é o conjunto de alternativas,  $d_i$  e  $t_i$  são a distância, o tempo, respectivamente, da cidade  $i$  até a cidade destino encontrada na solução  $\mathbf{A}$  e  $\bar{d}$  e  $\bar{t}$  são a distância e o tempo médios entre todas as cidades de cada alternativa  $A$ .

O critério binário é escrito como:

$$c_5(\mathbf{A}) = \begin{cases} 1, & \text{se } \exists t_i > 1.3 \quad \forall i \in [1, 250] \\ 0, & \text{cc.} \end{cases} \quad (12)$$

Outro elemento importante na Tomada de Decisão são as estruturas de preferências do DM, que definem os critérios prioritários na tomada de decisão e ajudam a ordenar as alternativas [9]. Estas definições iniciais são importantes e comuns aos três métodos de auxílio à tomada de decisão apresentados em seguida.

#### F. AHP

O primeiro método de auxílio a tomada de decisão a ser utilizado será o *Analytic Hierarchy Process* (AHP). O método consiste nos seguintes passos [10]:

- 1) Identificar os atributos importantes das alternativas.
- 2) Identificar a importância relativa entre os atributos, representando essa importância em uma escala de 1 a 9, em que 1 representa uma importância equivalente entre dois atributos e 9 representa uma importância máxima entre um atributo de uma solução  $i$  em relação a uma solução  $j$ .
- 3) Preencher uma matriz para cada critério do tamanho  $n \times n$ , em que  $n$  é o número de alternativas, contendo a importância relativa entre um atributo da linha  $i$  e o atributo da coluna  $j$ .
- 4) Avaliar cada matriz de acordo com seu auto-valor para verificar a coerência dos julgamentos. Caso o valor da incoerência seja muito elevado, deve-se repensar no

preenchimento das matrizes de importâncias relativas até obter um grau de incoerência aceitável.

- 5) Calcula-se os valores globais de preferência e ordena as alternativas.

Definidos os critérios citados na seção anterior, precisamos calcular as importâncias relativas entre eles. A escala do AHP define:

- 1 - Importância equivalente
- 3 - Importância moderada
- 5 - Importância forte
- 7 - Importância muito forte
- 9 - Importância extrema

Definimos que os critérios de menor distância e menor tempo são os mais importantes e equivalentes. O critério de tempo limite é o segundo mais importante na escala, já que foi definido para que o trabalhador tenha um mínimo de viagens por dia de trabalho. Assim, ele recebeu importância 3 em relação aos anteriores. Os critérios de desvio padrão foram definidos também equivalentes entre si e de importância 5 em relação aos primeiros.

A matriz que contém a informação das importâncias relativas e que é usada para o cálculo das prioridades será mostrada na seção III (Resultados).

Para que as relações sejam válidas é preciso que a inconsistência seja menor que 0.1. A inconsistência é calculada através da expressão:

$$IC = \frac{\lambda_{max} - n}{n - 1} \quad (13)$$

Onde  $\lambda_{max}$  é o maior auto-valor e  $n$  é o número de linhas da matriz de comparações  $n \times n$ .

As prioridades são calculadas através do **auto-vetor direito principal**  $v$  da matriz de comparações  $A$ :

$$Av = \lambda_{max}v \quad (14)$$

Os valores são depois normalizados.

As prioridades são calculadas para a matriz das relações entre os critérios e para as matrizes das relações entre as alternativas para cada critério.

Para definir estas últimas na escala de 1 a 9, foi feita uma normalização. Para cada critério, foram capturados o melhor e o pior valores. O melhor foi definido como 1 e o pior como 9, onde todos os outros tem valores intermediários referentes à normalização nesse intervalo.

Para definir as linhas das matrizes a partir dessa normalização, basta pegar todos os valores e dividi-los pelo valor da alternativa da linha em questão.

Por exemplo: se a normalização de 4 alternativas gerou (2.36 5.68 1 9), onde as alternativas estão na ordem que foram definidas inicialmente, a matriz passa a conter na primeira linha (2.36 5.68 1 9)/2.36 = (1 2.407 0.424 3.814), na segunda linha (2.36 5.68 1 9)/5.68 e assim sucessivamente.

Este processo foi utilizado para automatizar a definição dessas matrizes, visto que dependendo do número de alternativas e critérios, a quantidade de matrizes e seu tamanho

podem ser grandes problemas se feitos manualmente. Além disso, garantimos assim inconsistência nula.

Com as matrizes definidas, as prioridades são calculadas e por fim as prioridades globais de cada alternativa. Para isso basta pegar a prioridade da alternativa 1 no critério  $c_1$  multiplicada pela prioridade do critério  $c_1$ , depois somar com a prioridade no critério  $c_2$  multiplicada pela prioridade do critério  $c_2$ , e assim sucessivamente com todos os critérios e para cada alternativa.

Desta maneira teremos um ranking global onde as melhores alternativas dados nossos critérios aparecem com prioridades maiores.

## G. ELECTRE I

O segundo método utilizado foi o método *Elimination Et Choix Traduisant la Réalité* (ELECTRE, ou Eliminação e Escolha como Expressão da Realidade), mais precisamente o primeiro método da família, conhecido como ELECTRE I [11]. O passo a passo deste método é mostrado a seguir [12]:

- 1) Define os pesos  $w_j$  para cada um dos  $J = \{1, \dots, n_c\}$  critérios.
- 2) Estabeleça comparações entre cada par de alternativas, gerando três tipos de conjuntos:  $J^+$ , que consiste nas alternativas  $i$  cujo valor no critério  $c_j$ , com  $j \in J$ , são maiores que uma alternativa  $k$ ;  $J^-$ , que consiste nas alternativas  $i$  cujo valor no critério  $c_j$ , com  $j \in J$ , são menores que uma alternativa  $k$ ;  $J^=$ , que consiste nas alternativas  $i$  cujo valor no critério  $c_j$ , com  $j \in J$ , são iguais a uma alternativa  $k$ ;
- 3) Converta as relações entre ações em valores numéricos, sendo

$$P^+ = \sum_j w_j, \quad j \in J^+$$

$$P^= = \sum_j w_j, \quad j \in J^=$$

$$P^- = \sum_j w_j, \quad j \in J^-.$$

- 4) Calcule uma matriz de concordância, em que a concordância entre uma alternativa de índice  $i$  e uma alternativa de índice  $k$  é dada por

$$C_{ik} = \frac{P^+ + P^=}{\sum_{j \in J} w_j}.$$

- 5) Calcule a matriz de discordância, em que a discordância entre uma alternativa  $a_i$  e uma alternativa  $a_k$  é dada por

$$\begin{cases} 0, & \text{se } J^- = \phi \\ \delta_j \max(c_j(a_k) - c_j(a_i)), & j \in J^-, \quad \text{cc.} \end{cases}$$

- 6) Obtenha as relações de sobreclassificação:

$$a_i S a_k \iff \begin{cases} C_{ik}, \geq \tau_c \\ D_{ik}, \leq \tau_d, \end{cases}$$

em que  $\tau_c$  e  $\tau_d$  são os limiares de concordância e discordância, respectivamente, pré-definidos pelo DM.

Ao final do Electre I, espera-se obter uma matriz de sobreclassificação que posteriormente pode ser transformada em um grafo de dominância. As soluções ótimas podem ser encontradas no *kernel* do grafo, que consistem nas ações que não são sobreclassificadas por nenhuma outra ação, em conjunto das soluções incomparáveis, que não devem ser descartadas automaticamente.

## H. PROMETHEE II

O terceiro método utilizado foi o *Preference Ranking Organization Method for Enrichment Evaluations* (PROMETHEE), em que foi aplicada o segundo método da família (PROMETHEE II).

Os métodos PROMETHEE utilizam uma abordagem por fluxo num grafo de preferências. São definidas funções de preferência para cada par de alternativas, e as relações de preferência  $\{P, I, J\}$ , respectivamente "Preferência estrita", "Indiferença" e "Incomparabilidade" são determinadas através dos fluxos dessas funções de preferência [13].

No caso do PROMETHEE II, não há incomparabilidade, pois as preferências são definidas pelo fluxo total.

Os passos do método podem ser resumidos a:

- 1) Defina os pesos  $w_j$  para cada um dos  $J = \{1, \dots, n_c\}$  critérios.
- 2) Construa uma função de preferência. Pode ser de diversos tipos como: Quase-critério, Linear, Degrau, Trapezoidal, Gaussiana ou Usual, que é a mais simples e a que foi utilizada:

$$G(x) = \begin{cases} 0, & \text{se } x \leq 0 \\ 1, & \text{se } x > 0 \end{cases}$$

- 3) Estabeleça as relações de preferência por critério aplicando a função de preferência:

$$P_j(\mathbf{a}_i, \mathbf{a}_k) = G(c_j(\mathbf{a}_i) - c_j(\mathbf{a}_k))$$

Onde  $\mathbf{a}_i$  e  $\mathbf{a}_k$  se referem a duas alternativas presentes no grafo e  $c_j()$  ao valor que uma alternativa tem no critério  $j$ .

- 4) A partir das preferências por critério, calcule a preferência global:

$$P(\mathbf{a}_i, \mathbf{a}_k) = \frac{\sum_{j=1}^{n_c} w_j P_j(\mathbf{a}_i, \mathbf{a}_k)}{\sum_{j=1}^{n_c} w_j}$$

- 5) Calcule o fluxo líquido de cada alternativa ( $\mathbf{a}_i$ ) pela diferença entre o fluxo que entra e o fluxo que sai:

$$\phi(\mathbf{a}_i) = \phi^+(\mathbf{a}_i) - \phi^-(\mathbf{a}_i)$$

$$\phi(\mathbf{a}_i) = \sum_{\mathbf{a}_k \in A} [P(\mathbf{a}_i, \mathbf{a}_k) - P(\mathbf{a}_k, \mathbf{a}_i)]$$

- 6) Compare os fluxos e determine a relação de sobreclassificação final:

$$\mathbf{a}_i \text{ P } \mathbf{a}_k \iff \left\{ \text{se } \phi(\mathbf{a}_i) > \phi(\mathbf{a}_k) \right\}$$

$$\mathbf{a}_i \text{ I } \mathbf{a}_k \iff \left\{ \text{se } \phi(\mathbf{a}_i) = \phi(\mathbf{a}_k) \right\}$$

Com a relação de sobreclassificação definida, basta analisar quais alternativas têm mais preferência sobre as demais, o que pode ser ranqueado sem problemas, já que não há incomparabilidade. Assim temos as preferências finais das alternativas dados os critérios e seus pesos.

A próxima sessão deste artigo mostra os resultados obtidos utilizando os métodos acima descritos.

## III. RESULTADOS

O algoritmo de otimização multiobjetivo foi executado 5 vezes utilizando o método da Soma Ponderada. Depois, todas as soluções encontradas foram analisadas para definir as alternativas que seriam usadas na tomada de decisão, de forma a descartar as soluções dominadas para facilitar a tomada de decisão. A figura 2 mostra as onze soluções finais encontradas e seus resultados em relação aos dois critérios primários estabelecidos previamente. O número próximo à solução indica a posição dela no vetor de entrada e será utilizado para a identificação da solução.

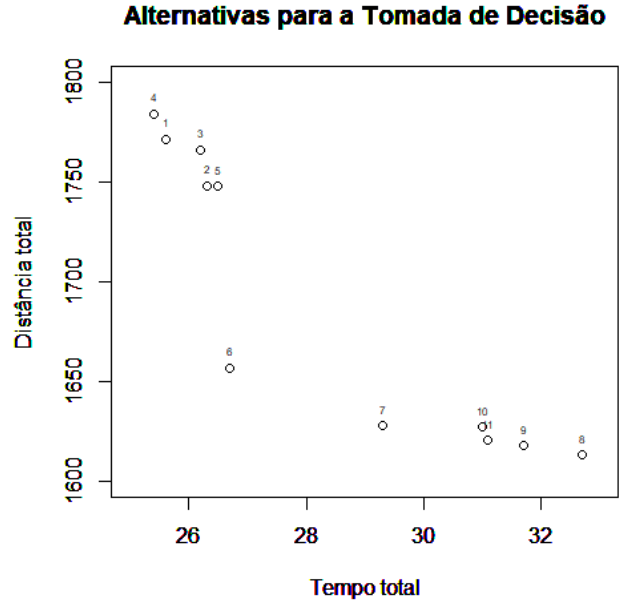


Fig. 2. Soluções Pareto Ótimas utilizadas na Tomada de Decisão.

Para a tomada de decisão, foi escolhido como objetivo final uma minimização dos cinco critérios propostos, propósito comum a todos os métodos aqui empregados. Esta minimização deve ocorrer ponderada pela ordem de preferência definida na sessão anterior deste artigo.

No método AHP, foram feitas 5 matrizes  $11 \times 11$ , uma para comparar cada uma das 11 alternativas em cada critério, além de uma matriz final  $5 \times 5$ , que comparava os critérios com o objetivo final. Esta matriz é mostrada na tabela II.

A inconsistência da matriz deve ser menor que 0.1 para que a relação entre os critérios seja válida. A inconsistência encontrada foi: **0.087**

TABLE II  
MATRIZ DE COMPARAÇÃO DOS CRITÉRIOS EM RELAÇÃO AO OBJETIVO

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
$c_1$	1.0	1.0	5.0	5.0	3.0
$c_2$	1.0	1.0	5.0	5.0	3.0
$c_3$	0.2	0.2	1.0	1.0	0.7
$c_4$	0.2	0.2	1.0	1.0	0.7
$c_5$	0.4	0.4	2.5	2.5	1.0

Os pesos obtidos através das prioridades do método do AHP, que posteriormente foram utilizados nos métodos ELECTRE I e PROMETHEE II, foram

$$\mathbf{w} = \begin{cases} w_1 = 0.35225230 \\ w_2 = 0.35225230 \\ w_3 = 0.07323682 \\ w_4 = 0.07323682 \\ w_5 = 0.14902177 \end{cases}$$

Ao fim da execução do método, as melhores alternativas foram ordenadas, na ordem da melhor para a pior, como:

$$1 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 7 \rightarrow 10 \rightarrow 2 \rightarrow 11 \rightarrow 6 \rightarrow 3 \rightarrow 5$$

Assim, a alternativa preferida após o método do AHP foi a alternativa de número 1.

Utilizando os pesos acima descritos, foi então executado o método ELECTRE I no conjunto de alternativas mostrado anteriormente na figura 2. O resultado da matriz de sobreclassificação é mostrado na tabela III, em que as linhas e as colunas representam as 11 alternativas propostas. Um número 1 em uma entrada da matriz demonstra uma relação de sobreclassificação da alternativa da linha  $i$  se comparada à alternativa da linha  $j$ , enquanto o número 0 indica que não há sobreclassificação. Como exemplo, é possível observar que a alternativa  $a_1$  sobreclassifica  $a_2, a_3, a_4$  e  $a_5$  e não sobreclassifica ela mesma (como esperado) além das alternativas  $a_6, a_7, a_8, a_9, a_{10}$  e  $a_{11}$  diretamente.

TABLE III  
MATRIZ DE SOBRECLASSIFICAÇÃO OBTIDA PELO MÉTODO ELECTRE I

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
$a_1$	0	1	1	1	1	0	0	0	0	0	0
$a_2$	0	0	0	0	0	0	0	0	0	0	0
$a_3$	0	0	0	0	1	0	0	0	0	0	0
$a_4$	0	1	0	0	0	0	0	0	0	0	0
$a_5$	0	0	0	0	0	0	0	0	0	0	0
$a_6$	0	0	1	0	1	0	0	0	0	0	1
$a_7$	0	0	0	0	0	0	0	0	0	0	0
$a_8$	0	0	0	0	0	0	0	0	0	0	0
$a_9$	0	0	0	0	0	0	0	0	0	0	0
$a_{10}$	0	0	0	0	0	0	0	1	1	0	0
$a_{11}$	0	0	0	0	0	0	0	0	0	0	0

A figura 3 mostra graficamente através de um grafo as sobreclassificações. É possível observar que o *kernel* do grafo, conforme a definição feita anteriormente neste artigo e conforme visto em [13], é composto pelas soluções  $\{1, 6, 7, 10\}$ . Como a alternativa 6 viola o critério  $c_5$ , que estabelecia que

o tempo entre as cidades deveria ser no máximo de  $1,3h$ , ela foi descartada. Entre as alternativas restantes, a escolhida foi a alternativa 1, mostrada no grafo em destaque, pois é a que apresentou menor desvio padrão em relação ao tempo e às distâncias em comparação às outras duas alternativas.

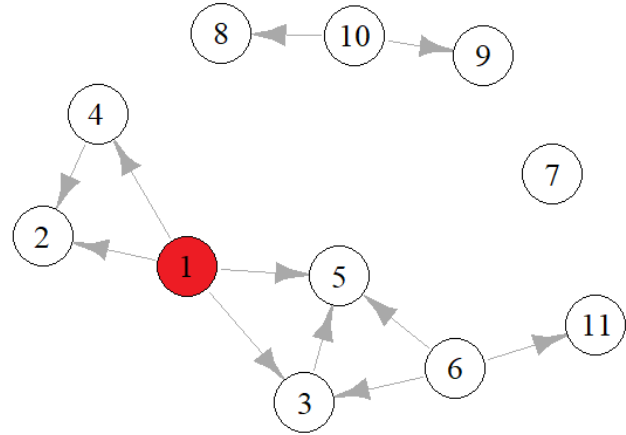


Fig. 3. Grafo obtido das relações de sobreclassificação após o ELECTRE I.

Por fim, assim como foi feito no método ELECTRE I, os pesos encontrados no AHP foram utilizados para a execução do método PROMETHEE II. A matriz de sobreclassificação encontrada é mostrada na tabela IV.

TABLE IV  
MATRIZ DE SOBRECLASSIFICAÇÃO OBTIDA PELO MÉTODO PROMETHEE II

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
$a_1$	0	1	1	1	1	1	1	1	1	1	1
$a_2$	0	0	1	0	1	0	1	1	1	0	1
$a_3$	0	0	0	0	1	0	1	0	0	0	1
$a_4$	0	1	1	0	1	1	1	1	1	1	1
$a_5$	0	0	0	0	0	0	0	0	0	0	1
$a_6$	0	1	1	0	1	0	1	1	1	0	1
$a_7$	0	0	0	0	1	0	0	0	0	0	1
$a_8$	0	0	1	0	1	0	1	0	0	0	1
$a_9$	0	0	1	0	1	0	1	0	0	0	1
$a_{10}$	0	1	1	0	1	1	1	1	1	1	1
$a_{11}$	0	0	0	0	0	0	0	0	0	0	0

Por fim, o grafo que representa a sobreclassificação das alternativas é mostrado na figura 4. Neste caso é fácil notar que o *kernel* do grafo corresponde apenas à alternativa 1, que aparece em destaque na figura e, portanto, essa é a alternativa escolhida nesta etapa.

Como os três métodos sugeriram a alternativa 1 como a preferida, os autores deste artigo, no papel de Tomadores de Decisão, escolheram esta alternativa como a melhor aproximação de uma possível solução ótima para o problema proposto. Os dados dessa alternativa são mostrados abaixo.

- Tempo total: 25,6h
- Distância total: 1771,6km
- $\sigma$  do tempo: 0,118h
- $\sigma$  da distância: 8,565km
- Tempo médio entre as cidades: 0,102h
- Distância média entre as cidades: 7,086km



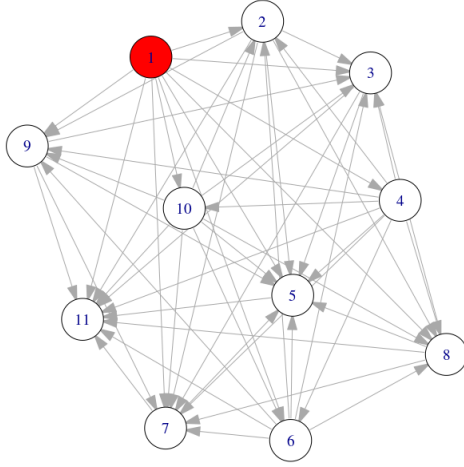


Fig. 4. Grafo obtido das relações de sobreclassificação após o PROMETHEE II.

A ordem das cidades a serem percorridas, de acordo com a alternativa escolhida, é mostrada a seguir. Cabe notar que o critério inicial do PCV, em que ele sai de uma cidade qualquer (aqui denominada de 1) e depois volta para a mesma cidade, sem que haja repetição, foi respeitado.

1 → 36 → 124 → 29 → 146 → 192 → 32 → 107 → 212 → 246 → 215 → 17 → 122 → 38 → 240 → 159 → 83 → 188 → 176 → 31 → 14 → 197 → 133 → 100 → 226 → 28 → 214 → 22 → 114 → 71 → 244 → 184 → 178 → 54 → 211 → 26 → 136 → 173 → 168 → 144 → 154 → 48 → 180 → 241 → 68 → 21 → 13 → 139 → 224 → 9 → 243 → 210 → 242 → 78 → 63 → 185 → 198 → 89 → 23 → 108 → 195 → 60 → 20 → 93 → 35 → 189 → 199 → 140 → 106 → 92 → 220 → 145 → 190 → 76 → 151 → 46 → 24 → 70 → 232 → 111 → 227 → 162 → 33 → 53 → 65 → 57 → 172 → 118 → 148 → 230 → 80 → 75 → 164 → 45 → 217 → 61 → 149 → 213 → 153 → 8 → 247 → 98 → 117 → 120 → 102 → 219 → 112 → 52 → 150 → 7 → 119 → 239 → 129 → 221 → 84 → 234 → 72 → 127 → 155 → 191 → 67 → 160 → 19 → 96 → 81 → 163 → 201 → 203 → 43 → 109 → 4 → 193 → 223 → 236 → 135 → 134 → 79 → 245 → 130 → 88 → 157 → 44 → 86 → 99 → 202 → 196 → 51 → 5 → 194 → 174 → 62 → 169 → 249 → 77 → 141 → 187 → 18 → 42 → 58 → 152 → 39 → 10 → 15 → 116 → 233 → 138 → 142 → 30 → 228 → 27 → 175 → 179 → 95 → 90 → 166 → 181 → 137 → 110 → 103 → 156 → 25 → 132 → 101 → 104 → 229 → 237 → 34 → 105 → 73 → 16 → 59 → 250 → 87 → 2 → 115 → 248 → 11 → 94 → 126 → 170 → 37 → 128 → 55 → 64 → 216 → 3 → 205 → 6 → 200 → 171 → 12 → 85 → 69 → 74 →

40 → 97 → 50 → 209 → 49 → 66 → 225 → 121 → 231 → 131 → 235 → 167 → 161 → 47 → 206 → 82 → 204 → 147 → 182 → 91 → 143 → 113 → 56 → 218 → 125 → 207 → 165 → 208 → 177 → 41 → 158 → 238 → 123 → 186 → 222 → 183 → 1.

#### IV. CONCLUSÃO

O Problema do Caixeiro Viajante é muito comum na literatura por ser um problema NP-completo, cuja solução em tempo polinomial implicaria na solução de todos os outros problemas NP-completos. Por não se saber se é possível essa solução exata, algoritmos de otimização são utilizados para aproximar soluções encontradas cada vez mais a um ótimo global.

O presente trabalho apresentou uma variante do PCV e otimizou uma solução inicial factível do problema utilizando o algoritmo *Simulated Annealing* Multi-objetivo. Em conjunto, foram implementados seis estruturas de vizinhanças com diferentes níveis de classificação cada uma. Em seguida, dentre as soluções Pareto-Ótimas encontradas, foi tomada uma decisão quanto à qual alternativa seria a preferida, utilizando os métodos AHP, ELECTRE I e PROMETHEE II. Com o auxílio dos métodos, os autores definiram uma solução final do problema.

Devido ao fato das alternativas usadas na tomada de decisão não corresponderem às soluções pareto-ótimas globais, uma vez que o método de otimização multicritério utilizado é baseado em uma metaheurística, a solução final não pode ser interpretada como uma solução ótima global, mas sim uma aproximação obtida por algoritmos de custo computacional relativamente baixo. Ainda assim, a solução apresentada satisfaz o critério binário estabelecido e minimizou os outros critérios, sendo uma solução mais viável que um caminho aleatório feito pelo Caixeiro Viajante.

Por fim, o uso de algoritmos de auxílio à tomada de decisão mostraram-se muito úteis para analisar as alternativas propostas e definir uma preferida de maneira mais robusta e com uma justificativa formal sólida. Esta ideia demonstra a importância dessa área de estudo, que pode ser aplicada em qualquer tipo de situação cuja tomada de decisão pode ser complexa demais para ser feita por análise humana das ações.

#### REFERÊNCIAS

- [1] F. D. M. Calado and A. P. Ladeira, "Problema do caixeiro viajante: Um estudo comparativo de técnicas de inteligência artificial," *e-xacta*, vol. 4, no. 1, 2011.
- [2] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM*, vol. 7, no. 4, pp. 326–329, 1960.
- [3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [4] K. Amine, "Multiobjective simulated annealing: Principles and algorithm variants," *Advances in Operations Research*, vol. 2019, pp. 1–13, 2019.
- [5] W. Ben-Ameur, "Computing the initial temperature of simulated annealing," *Computational Optimization and Applications*, vol. 29, no. 3, pp. 369–385, 2004.
- [6] D. Bertsimas and O. Nohadani, "Robust optimization with simulated annealing," *Journal of Global Optimization*, vol. 48, no. 2, pp. 323–334, 2009.

- [7] H. Shibata, "Pareto-optimality and gains-from-trade: A further elucidation," *Economica*, vol. 41, no. 161, p. 71, 1974.
- [8] K. Helsgaun, "General k-opt submoves for the lin-kernighan tsp heuristic," *Mathematical Programming Computation*, vol. 1, no. 2-3, pp. 119–163, 2009.
- [9] Y. Po-Lung, *Multiple-Criteria Decision Making*. New York, NY: Springer US, 1985.
- [10] T. L. Saaty, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, no. 1, p. 83, 2008.
- [11] J. R. Figueira, S. Greco, B. Roy, and R. Slowinski, "An overview of electre methods and their recent extensions," *Journal of Multi-Criteria Decision Analysis*, vol. 20, no. 1-2, pp. 61–85, 2012.
- [12] Y. Collette, *Multiobjective optimization*. Springer US, 2011.
- [13] J. Figueira, M. Ehrgott, and S. Greco, *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer US.