

Formulação matemática para o problema de orientação com seleção de hotéis e janelas de tempo

Cássio A. Rocha

Universidade Federal de Alagoas

Av. Lourival Melo Mota, S/n - Tabuleiro do Martins, Maceió - AL

car@ic.ufal.br

Alfredo Lima

Universidade Federal de Alagoas

Av. Lourival Melo Mota, S/n - Tabuleiro do Martins, Maceió - AL

alfredolms@ic.ufal.br

Rian Gabriel Santos Pinheiro

Universidade Federal de Alagoas

Av. Lourival Melo Mota, S/n - Tabuleiro do Martins, Maceió - AL

rian@ic.ufal.br

Bruno Nogueira

Universidade Federal de Alagoas

Av. Lourival Melo Mota, S/n - Tabuleiro do Martins, Maceió - AL

Universidade de Pernambuco

Programa de Pós-graduação em Engenharia de Computação, Recife - PE

bruno@ic.ufal.br

RESUMO

Este artigo propõe um modelo de programação inteira para o Problema de Orientação considerando restrições realísticas para uma aplicação de geração de rotas turísticas. Mais especificamente, consideramos o problema chamado Problema da Orientação com Seleção de Hotel e Janelas de Tempo (OPHS-TW). Nosso modelo para o OPHS-TW foi testado em 395 instâncias da literatura. Os resultados experimentais mostram que ele foi capaz de encontrar soluções previamente desconhecidas para 32 das 395 instâncias da literatura. Também foi capaz de comprovar a otimalidade de um total de 360 soluções, incluindo todas as soluções previamente desconhecidas.

PALAVRAS CHAVE. Problema de Orientação, Problema de Design de Viagem Turística, Itinerários Turísticos

Tópicos PM – Programação Matemática, OC – Otimização Combinatória

ABSTRACT

This paper presents an integer programming formulation for the Orienteering Problem with Hotel Selection and Time Windows (OPHS-TW). To the best of the authors' knowledge, this is the first ILP-based approach to the problem. Our devised formulation discovered previously

unknown solutions for 32 out of 395 instances from the literature. Furthermore, it was able to demonstrate the optimality of a total of 360 solutions, encompassing all formerly unknown alternatives.

KEYWORDS. Orienteering Problem. Tourist Trip Design Problem. Tourist itineraries

Paper topics PM – Mathematical Programming, OC – Combinatorial Optimization

1. Introdução

As tecnologias se tornaram fortes aliadas para o crescimento do turismo. Exemplos dessas tecnologias incluem reservas online, como Booking e Trivago, que auxiliam no planejamento de uma viagem turística. Os sistemas de reserva online facilitam a busca por quartos de hotel disponíveis ou assentos de voo. No entanto, os turistas ainda gastam muito tempo planejando itinerários turísticos, pois frequentemente precisam tomar decisões difíceis sobre os melhores itinerários entre os pontos turísticos que consideram atrativos. Vários fatores precisam ser considerados nesse problema de planejamento de viagem, como tempo de viagem de um ponto a outro, horários de funcionamento e ordem de visita. De fato, esse problema é conhecido por ser $\mathcal{NP}$ -difícil e, portanto, os turistas precisam de uma ferramenta que os livre de resolvê-lo manualmente.

Os pioneiros no campo da otimização de itinerários turísticos foram Vansteenwegen e Van Oudheusden [2007]. Seus estudos se concentraram em analisar como problemas de roteamento com lucro podem aumentar a satisfação dos turistas na geração de rotas. Problemas de roteamento com lucro, de acordo com Archetti et al. [2017], têm como decisão crucial a identificação de um subconjunto de clientes “convenientes” a serem atendidos, ao contrário do que acontece no roteamento clássico, em que todos os clientes devem ser atendidos. Implementações práticas desses problemas para otimizar itinerários turísticos foram realizadas em Singapura por Gunawan e Lau [2016] e em Atenas, Grécia, por Gunawan et al. [2017].

Problemas de roteamento com lucro são tradicionalmente classificados em duas categorias: (i) problemas de roteamento com lucro nos vértices, nos quais os clientes estão localizados nos vértices de um grafo, e (ii) problemas de roteamento com lucro nos arcos, nos quais os clientes correspondem às arestas do grafo. O primeiro pode ser dividido ainda em três classes principais [Archetti et al., 2017]:

Orienteering: A função objetivo é maximizar o lucro total coletado. Nesse caso, é imposto uma restrição de duração máxima ou uma restrição de capacidade (ou ambas) nas rotas do veículo. O problema mais estudado nessa classe é o Problema de Orientação.

Prize-Collecting: A função objetivo é minimizar o custo total de viagem. Nesse caso, é imposto uma restrição para coletar um valor mínimo de lucro.

Profitable Tour: A função objetivo maximiza a diferença entre o lucro total coletado e o custo de viagem.

O problema abordado neste trabalho pertence à categoria de Orientação (Orienteering), mas também inclui fatores como restrições de janelas de tempo, seleção de hotel e rotas múltiplas. Na literatura, essa variante do problema de Orientação é conhecida como Problema de Orientação com Seleção de Hotel e Janelas de Tempo (OPHS-TW) [Divsalar et al., 2014b]. O OPHS-TW pode ser definido formalmente da seguinte forma. Seja N um conjunto de pontos de interesse (POI), H um conjunto de hotéis, NH a união de N e H ($NH = N \cup H$), onde $n = |N|$, $h = |H|$, $nh = |NH|$, e m é o número de dias da viagem. Para cada POI $i \in N$, consideramos que S_i é a pontuação, T_i é o tempo de serviço, O_i e C_i são funções que indicam o horário de abertura e fechamento. Além disso, T_d indica um orçamento de tempo para cada dia $d \in m$, E é um conjunto de links ($E \subseteq NH \times NH$), e L_{ij} é uma função que indica o tempo do caminho de i para j , onde $(i, j) \in E$. Com base nessas definições, o objetivo do OPHS-TW é encontrar um subconjunto de N , um subconjunto de H e uma viagem com m rotas conectadas, de modo que cada dia comece e termine em um hotel e a pontuação da viagem seja maximizada. Observa-se que cada plano de

viagem diária é chamado de rota, e a próxima rota deve começar no mesmo hotel onde a rota atual terminou.

Este trabalho propõe um modelo exato do OPHS-TW baseado em Programação Linear Inteira (ILP). Foram realizados experimentos computacionais em 395 instâncias encontradas na literatura. Os resultados experimentais mostram que nosso método exato pode comprovar diversos ótimos previamente desconhecidos. Em particular, nosso algoritmo encontrou 32 soluções desconhecidas na literatura. Todas as soluções desconhecidas foram comprovadas como ótimas. No total, 360 instâncias foram resolvidas de forma ótima.

Na próxima seção, forneceremos uma visão geral da literatura, que estabelece o contexto para a formulação matemática proposta introduzida na Seção 3. Em seguida, na Seção 4, apresentamos os resultados experimentais. Por fim, na Seção 5, discutimos nossas conclusões e trabalhos futuros.

2. Trabalhos relacionados

A Figura 1 ilustra as variantes mais proeminentes do Problema de Orientação (OP, do inglês Orienteering Problem), e a Tabela 1 descreve cada uma dessas variantes. Vale ressaltar que o Problema do Caixeiro Viajante (TSP, do inglês Traveling Salesman Problem) e o OP possuem variantes semelhantes, como mostrado na Figura 2. As diferenças principais entre o TSP e o OP são que no TSP todos os vértices devem ser visitados e o objetivo é minimizar o tempo total da viagem. No entanto, no OP nem todos os vértices precisam ser visitados e a função objetivo é maximizar a pontuação total. As variantes do TSP e do OP possuem siglas comuns, como “TW”, indicando que o modelo possui janelas de tempo, e “HS”, indicando que há seleção de hotel no modelo. No OP, o acréscimo da sigla “T” indica que o modelo possui equipes competindo entre si. Por exemplo, no Problema de Orientação em Equipes (TOP, do inglês Team Orienteering Problem), cada membro de uma equipe é considerado uma rota, e a equipe é considerada um conjunto de rotas (viagem). Além disso, no OP, um fator distintivo entre um modelo com “HS” em seu nome e um modelo com “T” é que o primeiro incorpora tanto opções de rotas múltiplas quanto seleção de hotel, enquanto o último apenas indica a presença de rotas múltiplas. De acordo com Divsalar et al. [2013], a principal diferença entre o TOP e o OPHS é que no TOP todas as rotas devem começar e terminar no mesmo vértice e nenhum hotel precisa ser selecionado.

Figura 1: Variantes do OP

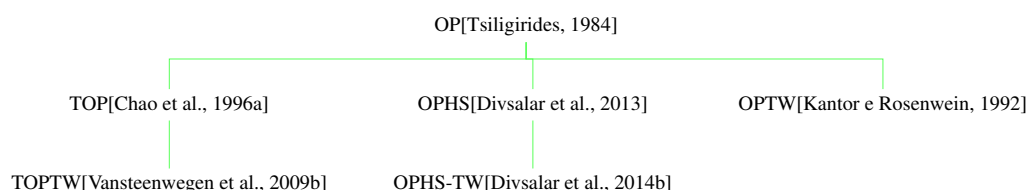


Figura 2: Variantes do TSP

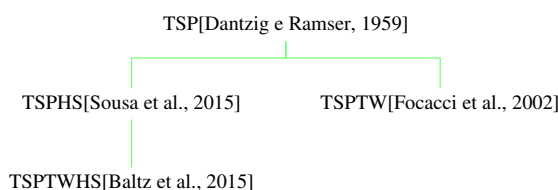


Tabela 1: OP e suas variantes iniciais.

Nome	Autor	Descrição
OP	Tsiligirides [1984]	A função objetivo é maximizar a pontuação total. Seleção de POIs para um dia.
TOP	Chao et al. [1996a]	Seleção de POIs para vários dias.
OPTW	Kantor e Rosenwein [1992]	Seleção de POIs para um dia com janelas de tempo.
TOPTW	Vansteenwegen et al. [2009b]	Seleção de POIs para vários dias com janelas de tempo.
OPHS	Divsalar et al. [2013]	Seleção de hotel e POIs para um ou vários dias.
OPHS-TW	Divsalar et al. [2014b]	Seleção de hotel e POIs para um ou vários dias com janelas de tempo.

O OP foi inicialmente publicado por Tsiligirides [1984]. Sua construção foi baseada em um esporte de orientação que combina corrida em campo aberto e navegação em floresta, utilizando mapa e bússola. Os competidores iniciam em intervalos de tempo e têm como objetivo encontrar vários “pontos de controle” que foram colocados na floresta e cujas localizações estão marcadas nos mapas dos competidores. Eles não precisam visitar todos os pontos. Cada ponto de controle possui uma pontuação, e o objetivo do competidor é maximizar sua pontuação total dentro do limite de tempo da competição. Portanto, cada competidor deve planejar sua rota para acumular a pontuação mais alta com tempo suficiente para retornar à linha de chegada.

O OP pode ser modelado como um problema de otimização multinível, conforme descrito por Chao et al. [1996a]. No primeiro nível, a tarefa é escolher um subconjunto de pontos de controle a serem visitados. No segundo nível, é necessário resolver um Problema do Caixeiro Viajante (PCV) sobre o subconjunto selecionado de pontos de controle. Portanto, o OP se enquadra na classe de problemas $\mathcal{NP}$ -difíceis, pois contém o conhecido PCV como um caso especial.

Após a publicação inicial do OP, surgiram várias aplicações e variantes. Chao et al. [1996b] propuseram uma variante conhecida como Problema de Orientação em Equipe (TOP), que estende a versão de um único competidor do esporte. Nessa variante, uma equipe composta por vários competidores inicia no mesmo ponto. Cada membro da equipe tenta visitar o maior número possível de pontos de controle dentro de um limite de tempo determinado e termina no ponto de chegada. A equipe com a pontuação mais alta vence. Vansteenwegen et al. [2009a] propuseram um modelo matemático e uma meta-heurística de busca local guiada para resolver o TOP.

Vansteenwegen et al. [2009b] formularam a variante Problema de Orientação em Equipe com Janelas de Tempo (TOPTW) para gerar itinerários. No TOPTW, é fornecido um conjunto de locais, cada um com uma pontuação, tempo de serviço e janela de tempo. O objetivo é maximizar a soma das pontuações coletadas por um número fixo de rotas. Cada rota pode ser interpretada como uma viagem diária com tempo limitado de visitação.

Divsalar et al. [2013] sugeriram o Problema de Orientação com Seleção de Hotel (OPHS), no qual é fornecido um conjunto de POIs com pontuação e um conjunto de hotéis. O objetivo é determinar um número fixo de rotas conectadas que visitam alguns vértices e maximizam a soma das pontuações coletadas. Cada rota tem uma duração limitada e deve começar e terminar em um

dos hotéis. Os hotéis não possuem pontuações. O início e o fim da viagem estão definidos pelo hotel de chegada. Divsalar et al. [2014a] também implementaram um algoritmo memético para o OPHS, que consiste em dois níveis: no primeiro nível, um componente genético concentra-se em encontrar uma boa sequência de hotéis intermediários, enquanto no segundo nível, seis movimentos de busca local incorporados em uma estrutura de vizinhança variável lidam com a seleção e sequenciamento de POIs entre hotéis.

O OPHS-TW foi inicialmente proposto por Divsalar et al. [2014b]. A cada POI é atribuída uma janela de tempo. Na prática, o POI possui horários de visita limitados. Portanto, os autores adicionaram um horário de abertura e fechamento para cada POI. Em sua formulação, o tempo gasto (tempo de serviço) em cada POI não é considerado, o que é considerado uma desvantagem para o modelo, uma vez que um problema real deve levar em conta o tempo gasto em cada atração turística. Eles propuseram um Algoritmo Genético Híbrido com Descida de Vizinhança Variável (GA-VND) como meta-heurística. Os autores adaptaram instâncias do OPHS para incluir janelas de tempo. Dessa forma, eles consideraram que uma solução ótima para a instância do OPHS permanece ótima para a instância do OPHS-TW, desde que as restrições da janela de tempo sejam respeitadas.

Os problemas que mais se assemelham ao problema abordado neste trabalho, OPHS-TW, são OPHS [Divsalar et al., 2013], TOPTW [Vansteenwegen et al., 2009b] e TSP com Múltiplas Janelas de Tempo e Seleção de Hotel (TSPTWHS) [Baltz et al., 2015]. No entanto, OPHS não considera as janelas de tempo, TOPTW não considera hotéis e, no TSPTWHS, a função objetivo não considera a seleção de vértices por pontuação. Em relação aos métodos para resolver o OPHS-TW, existem duas meta-heurísticas: (i) a heurística de três componentes (I3CH) proposta por Hu e Lim [2014] e adaptada para o OPHS-TW por Cheung [2016], e (ii) o método híbrido GA-VND proposto por Divsalar et al. [2014b]. Até onde os autores têm conhecimento, este trabalho é o primeiro a propor um método exato para o OPHS-TW.

3. Formulação matemática

A formulação matemática do OPHS-TW é ilustrada abaixo. Algumas das restrições propostas foram baseadas no modelo TOPTW desenvolvido por Vansteenwegen et al. [2009b]. As variáveis de decisão são:

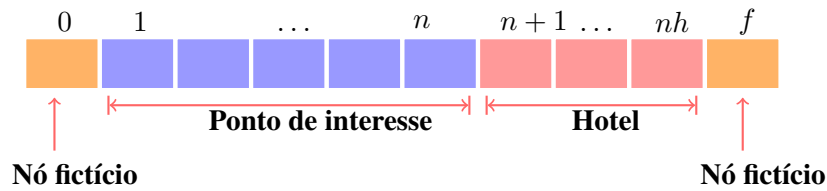
- $x_{ij}^d = 1$ se, na rota d , uma visita à localização i é seguida por uma visita à localização j , 0 caso contrário. $\forall : i, j \in \{1, \dots, nh + 1\}, d \in \{1, \dots, m\}$;
- $y_{id} = 1$ se a localização i é visitada na rota d , 0 caso contrário. $\forall : i \in \{1, \dots, nh + 1\}, d \in \{1, \dots, m\}$;
- s_{id} é o início do serviço no local i na rota d , $\forall : i \in \{1, \dots, nh + 1\}, d \in \{1, \dots, m\}$.

A Figura 3 exemplifica os índices i, j utilizados na variável de decisão x_{ij}^d , que possui *nós fictícios*, POIs (Pontos de Interesse) e hotéis. No total, são utilizados dois *nós fictícios*, representados pelos índices 0 e $nh + 1$ (iguais a f). Eles são usados apenas para facilitar a construção da seleção de hotéis. Os POIs são obtidos nos índices $1, \dots, n$. Os hotéis estão nos índices $n + 1, \dots, nh$ (iguais a $n + h$). Os parâmetros utilizados são:

- $S_i \rightarrow$ pontuação do POI i ;
- $T_i \rightarrow$ tempo de visita (ou tempo de serviço) do POI i ;
- $T_d \rightarrow$ tempo da rota do dia d ;

- $L_{ij} \rightarrow$ tempo de deslocamento da localização i para j ;
- $O_i \rightarrow$ horário de início do serviço i ;
- $C_i \rightarrow$ horário em que o serviço i fecha.

Figura 3: Descrição das variáveis de decisão x_{ij}^d



A função objetivo (1) maximiza o total de pontos coletados. As restrições (2) garantem que um POI seja visitado no máximo uma vez. As restrições (3) e (4) determinam a linha do tempo e a conectividade de cada rota, onde M é uma constante grande. As restrições (5) garantem que exista uma aresta do nó fictício para um hotel, e as restrições (6) garantem que exista uma aresta do hotel para o nó fictício. As restrições (7) limitam o tempo de rota para cada dia, e as restrições (8) garantem que o hotel em que o turista dorme seja o mesmo em que ele acorda. As restrições (9) e (10) garantem que todos os dias um turista sai de um hotel e retorna ao hotel. A restrição (11) define o primeiro hotel como o inicial do conjunto. A restrição (12) define o segundo hotel como o final do conjunto. As restrições (11) e (12) podem ser removidas se os hotéis inicial e final não forem fixos. As restrições (13) e (14) garantem que uma visita a uma atração seja iniciada apenas dentro da janela de tempo. As restrições (15) e (16) garantem que não haja uma aresta do nó fictício para uma atração. As variáveis de decisão são dadas por (17), (18) e (19).

$$\max \sum_{d=1}^m \sum_{i=1}^n S_i y_{id} \quad (1)$$

$$\text{s.t.} \quad \sum_{d=1}^m y_{kd} \leq 1 \quad \forall k \in \{1, \dots, n\} \quad (2)$$

$$s_{id} + T_i + L_{ij} - s_{jd} \leq M(1 - x_{ij}^d) \quad \forall i, j \in \{0, \dots, nh\}, d \in \{1, \dots, m\} \quad (3)$$

($i \neq j$), ($n+1 > i > nh$), ($n+1 > j > nh$)

$$\sum_{i=0}^{nh} x_{ik}^d = \sum_{j=1}^{nh} (x_{kj}^d + x_{kf}^d) = y_{kd} \quad \forall k \in \{1, \dots, nh\}, d \in \{1, \dots, m\} \quad (4)$$

$$\sum_{j=n+1}^{nh} x_{0j}^d = 1 \quad \forall d \in \{1, \dots, m\} \quad (5)$$

$$\sum_{j=n+1}^{nh} x_{jf}^d = 1 \quad \forall d \in \{1, \dots, m\} \quad (6)$$

$$\sum_{i=0}^{nh} \left(T_i y_{id} + \sum_{j=1}^f L_{ij} x_{ij}^d \right) \leq T_d \quad \forall d \in \{1, \dots, m\} \quad (7)$$

$$x_{if}^d = x_{0i}^{(d+1)} \quad \forall d \in \{1, \dots, m-1\}, i \in \{n+1, \dots, nh\} \quad (8)$$

$$\sum_{i=1}^n \sum_{j=n+1}^{nh} x_{ij}^d = 1 \quad \forall d \in \{1, \dots, m\} \quad (9)$$

$$\sum_{i=1}^n \sum_{j=n+1}^{nh} x_{ji}^d = 1 \quad \forall d \in \{1, \dots, m\} \quad (10)$$

$$x_{0(n+1)}^1 = 1 \quad (11)$$

$$x_{(n+2)f}^m = 1 \quad (12)$$

$$O_i \leq s_{id} \quad \forall i \in \{1, \dots, n\}, d \in \{1, \dots, m\} \quad (13)$$

$$s_{id} \leq C_i \quad \forall i \in \{1, \dots, n\}, d \in \{1, \dots, m\} \quad (14)$$

$$x_{if}^d = 0 \quad \forall d \in \{1, \dots, m\}, i \in \{1, \dots, n\} \quad (15)$$

$$x_{0i}^d = 0 \quad \forall d \in \{1, \dots, m\}, i \in \{1, \dots, n\} \quad (16)$$

$$x_{ij}^d \in \{0, 1\} \quad \forall i, j \in \{1, \dots, nh+1\}, d \in \{1, \dots, m\} \quad (17)$$

$$0 \leq y_{id} \leq 1 \quad \forall i \in \{1, \dots, nh+1\}, d \in \{1, \dots, m\} \quad (18)$$

$$s_{id} \geq 0 \quad \forall i \in \{1, \dots, nh+1\}, d \in \{1, \dots, m\} \quad (19)$$

Para tornar o modelo mais robusto, a constante M nas Restrições (3) foi definida da seguinte forma:

$$M_{ij} = \begin{cases} 1 + C_i + T_i + L_{ij} - O_j, & \text{se } i \in \{1, \dots, n\} \text{ e } j \in \{1, \dots, n\} \\ T_{max} + L_{ij}, & \text{caso contrário,} \end{cases} \quad (20)$$

observe que no primeiro caso ($i \in \{1, \dots, n\}$ e $j \in \{1, \dots, n\}$), i e j correspondem a dois POIs, enquanto no segundo caso, i ou j é um hotel. Através dessa definição, é possível remover variáveis do modelo quando $M_{ij} < 0$, pois isso indica que o ponto j não pode ser visitado após o ponto i . Assim, quando $M_{ij} < 0$, é adicionada a restrição $x_{ij}^d = 0, \forall d \in \{1, \dots, m\}$.

4. Experimentos computacionais

Esta seção avalia os resultados do método exato proposto. Nos nossos experimentos, utilizamos o software de otimização linear CPLEX versão 12.9. Os experimentos foram realizados em um sistema com processador Intel Xeon(R) CPU E5-2670 0 @ 2.60GHz × 24; Memória 100 GiB; Linux Mint Release Linux Mint 20.3 Una 64-bit, Kernel Linux 5.4.0-94-generic x86_64.

Os resultados foram obtidos a partir de 16 conjuntos de instâncias do OPHS-TW¹, fornecidos por Divsalar et al. [2014b]. Cada conjunto de instâncias compartilha características comuns,

¹<http://www.mech.kuleuven.be/en/cib/op>

como o mesmo número de hotéis intermediários e dias. A Tabela 2 detalha cada conjunto de instâncias. Nesta tabela, ‘Conj.’ lista os 16 conjuntos de instâncias, ‘Min’ e ‘Max’ são, respectivamente, o número mínimo e máximo de POIs no conjunto, ‘Hotéis intermediários’ é o número de hotéis que fazem conexões entre os dias da viagem (excluindo o hotel inicial e final), ‘Dias’ é o número total de dias da viagem, ‘Total’ é o número total de instâncias por conjunto. Na parte inferior da tabela, é mostrado o número total de instâncias de todos os conjuntos. Além disso, indicamos na coluna ‘Inconsistências’ o número de instâncias que apresentaram alguma inconsistência e não puderam ser avaliadas, por exemplo, POIs em que o horário de abertura é maior que o horário de fechamento. Nos experimentos computacionais, definimos o hotel inicial como o primeiro do conjunto e o hotel final como o segundo, para ser semelhante ao trabalho de Divsalar et al. [2014b].

Tabela 2: Descrição do conjunto de instâncias de Divsalar et al. [2014b]

Conf.	POIs		Hotéis intermediários	Dias	Total	Inconsistências
	Min	Max				
1	31	101	1	2	35	2
2	31	101	2	3	35	0
3	31	101	5	3	35	1
4	31	101	3	4	35	1
5	31	101	6	4	35	0
6	63	99	10	4	22	2
7	63	99	10	5	22	2
8	63	99	10	6	22	3
9	63	99	12	4	22	2
10	63	99	12	5	22	0
11	63	99	12	6	22	1
12	63	99	15	4	22	0
13	63	99	15	5	22	0
14	63	99	15	6	22	2
15	99	99	15	8	13	5
16	99	99	15	10	9	1
Total					395	22

A abordagem proposta foi comparada com duas heurísticas para o problema: (i) GA-VND de Divsalar et al. [2014b] e (ii) I3CH de Hu e Lim [2014], que foi adaptado para o OPHS-TW por Cheung [2016]. Um resumo dos resultados dessa comparação é apresentado na Tabela 3 (resultados detalhados completos disponíveis aqui²). Como não conseguimos obter os códigos-fonte do GA-VND e do I3CH, foram replicados os resultados que aparecem em Divsalar et al. [2014b] e Cheung [2016]. Observamos que os autores do GA-VND mencionam que seus experimentos foram realizados em um desktop Intel Core 2 com processador de 3,00 GHz e 4,00 GB de RAM, enquanto os autores do I3CH realizaram seus experimentos em um Intel Core i7-6500U CPU com clock de 2,50 GHz e 8 GB de RAM. Na Tabela 3, a coluna ‘Total’ representa o número de instâncias avaliadas no conjunto (excluindo as instâncias com inconsistências), ‘Melhor’ é o número de instâncias em que um método encontrou a melhor solução conhecida, ‘T(s)’ é o tempo médio de CPU para cada conjunto, ‘Ótimo’ é o número de instâncias em que um método provou a otimalidade e ‘GAP’ é a

²REMOVIDO DEVIDO AO BLIND REVIEW

diferença média para um conjunto de instâncias (valor retornado pelo MIP solver).

Tabela 3: Comparação da abordagem exata proposta com GA-VND e I3CH

Conj.	Total	GA-VND		I3CH		Exato			
		Melhor	T(s)	Melhor	T(s)	Melhor	Ótimo	GAP	T(s)
1	33	28	0,31	19	209,58	33	33	0,00	11,42
2	35	32	0,44	12	161,65	35	35	0,00	9,92
3	34	30	0,87			34	34	0,00	22,19
4	34	33	0,69	1	135,11	34	34	0,00	202,00
5	35	31	2,16			35	35	0,00	144,51
6	20	19	46,15			19	19	0,96	4246,86
7	20	16	115,95			19	19	0,67	3975,06
8	19	16	148,59			18	18	0,57	1315,25
9	20	18	71,04			20	20	0,00	2382,84
10	22	19	196,95			21	21	0,52	5021,39
11	21	18	265,02			19	19	1,18	4911,53
12	22	20	179,91			21	20	0,21	3998,65
13	22	22	396,36			21	21	0,18	3994,75
14	20	18	471,98			20	20	0,00	1896,81
15	8	7	975,92			7	7	1,38	6238,27
16	8	8	900,11			5	5	4,28	13681,86
Média			235,78		168,78			0,62	3253,33
Total	373	335		32		361	360		

A Tabela 3 mostra que o método Exato proposto conseguiu com sucesso comprovar a otimalidade de um grande número de instâncias. Especificamente, das 373 instâncias testadas, o método Exato provou a otimalidade de 360 soluções. Além disso, identificou 32 melhores soluções desconhecidas previamente, todas as quais foram posteriormente comprovadas como ótimas. Os valores de GAP demonstram que, mesmo quando nosso método não consegue provar a otimalidade, ele encontra boas soluções.

Embora os autores do I3CH não tenham testado seu algoritmo em todos os conjuntos de instâncias, os resultados disponíveis indicam que o GA-VND obtém um desempenho melhor que o I3CH. Ao comparar os resultados do GA-VND com nossa abordagem, observamos que nosso método foi capaz de obter 38 soluções que não foram obtidas pelo GA-VND. Por outro lado, o GA-VND conseguiu encontrar 12 soluções diferentes que não eram alcançáveis com nosso método. Essas descobertas sugerem que nosso método pode obter resultados bastante competitivos em instâncias de pequeno a médio porte.

5. Conclusão

Até onde se sabe, este artigo apresentou a primeira metodologia exata para o problema OPHS-TW, que visa gerar rotas turísticas considerando restrições do mundo real. Os experimentos computacionais realizados em instâncias do OPHS-TW encontradas na literatura mostram que nosso método comprovou muitos ótimos previamente desconhecidos. Uma vez que as soluções ótimas para essas instâncias são conhecidas, elas podem ser usadas como referência para pesquisas

futuras. Os resultados experimentais também indicaram que nosso método pode obter resultados competitivos em comparação com as meta-heurísticas de ponta para o problema. Para trabalhos futuros, planejamos desenvolver uma nova meta-heurística para o OPHS-TW. Também estamos considerando adicionar restrições mais realistas, como janelas de tempo múltiplas, já que alguns POIs têm várias aberturas e fechamentos.

Referências

- Archetti, C., Bertazzi, L., Laganà, D., e Vocaturo, F. (2017). The undirected capacitated general routing problem with profits. *European Journal of Operational Research*, 257(3):822–833.
- Baltz, A., El Ouali, M., Jäger, G., Sauerland, V., e Srivastav, A. (2015). Exact and heuristic algorithms for the travelling salesman problem with multiple time windows and hotel selection. *Journal of the Operational Research Society*, 66(4):615–626.
- Chao, I.-M., Golden, B. L., e Wasil, E. A. (1996a). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, 88(3):475 – 489. ISSN 0377-2217. URL <http://www.sciencedirect.com/science/article/pii/0377221795000356>.
- Chao, I.-M., Golden, B. L., e Wasil, E. A. (1996b). The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/0377221794002894>.
- Cheung, Y. (2016). Implementation of the iterative three-component heuristic for the team orienteering problem with time windows. *Erasmus University Rotterdam*. URL <http://hdl.handle.net/2105/34302>.
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1): 80–91.
- Divsalar, A., Vansteenwegen, P., e Cattrysse, D. (2013). A variable neighborhood search method for the orienteering problem with hotel selection. *International Journal of Production Economics*, 145(1):150–160. ISSN 0925-5273. URL <https://www.sciencedirect.com/science/article/pii/S0925527313000285>.
- Divsalar, A., Vansteenwegen, P., Sörensen, K., e Cattrysse, D. (2014a). A memetic algorithm for the orienteering problem with hotel selection. *European Journal of Operational Research*, 237(1):29–49. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/S0377221714000034>.
- Divsalar, A., Vansteenwegen, P., Chitsaz, M., Sörensen, K., e Cattrysse, D. (2014b). Personalized multi-day trips to touristic regions: A hybrid ga-vnd approach. In Blum, C. e Ochoa, G., editors, *Evolutionary Computation in Combinatorial Optimisation*, p. 194–205, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-662-44320-0.
- Focacci, F., Lodi, A., e Milano, M. (2002). A hybrid exact algorithm for the tsptw. *INFORMS journal on Computing*, 14(4):403–417.

- Gunawan, A. e Lau, H. C. (2016). A fast algorithm for personalized travel planning recommendation. PATAT.
- Gunawan, A., Lau, H. C., Vansteenwegen, P., e Lu, K. (2017). Well-tuned algorithms for the team orienteering problem with time windows. *Journal of the Operational Research Society*, 68(8): 861–876.
- Hu, Q. e Lim, A. (2014). An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 232(2):276–286.
- Kantor, M. G. e Rosenwein, M. B. (1992). The orienteering problem with time windows. *Journal of the Operational Research Society*, 43(6):629–635.
- Sousa, M. M., Ochi, L. S., Coelho, I. M., e Gonçalves, L. B. (2015). A variable neighborhood search heuristic for the traveling salesman problem with hotel selection. In *2015 Latin American Computing Conference (CLEI)*, p. 1–12. IEEE.
- Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809.
- Vansteenwegen, P., Souffriau, W., Berghe, G. V., e Van Oudheusden, D. (2009a). A guided local search metaheuristic for the team orienteering problem. *European journal of operational research*, 196(1):118–127.
- Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., e Van Oudheusden, D. (2009b). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12):3281 – 3290. ISSN 0305-0548. URL <http://www.sciencedirect.com/science/article/pii/S030505480900080X>. New developments on hub location.
- Vansteenwegen, P. e Van Oudheusden, D. (2007). The mobile tourist guide: an or opportunity. *OR insight*, 20(3):21–27.