



ALGORITMO GENÉTICO BASEADO EM CHAVES ALEATÓRIAS PARA O *ORIENTEERING PROBLEM WITH TIME WINDOWS*

Vinícius Almeida Gonçalves

Daniel Morais dos Reis

Eduardo Habib Bechelane Maia

Breno Alves Beirigo

vinicius.algo@gmail.com

daniel.morais@gmail.com

eduardohabib@gmail.com

brenobeirigo@gmail.com

Centro Federal de Educação Tecnológica de Minas Gerais – CEFET MG

Rua Álvares de Azevedo 400 - Bela Vista - CEP 35503- 822 - Divinópolis - MG- Brasil

Sérgio Ricardo de Souza

Anolan Yamilé Milanés Barrientos

sergio@dppg.cefetmg.br

anolan@decom.cefetmg.br

Centro Federal de Educação Tecnológica de Minas Gerais – CEFET MG

Av. Amazonas 7675 - Nova Gameleira- CEP 30510-000 - Belo Horizonte - MG - Brasil

Abstract. *The Trip Design Problem (TTDP) is a computational problem related to path planning among points of interests in a touristic area. It can be modeled as a well-known routing problem called Orienteering Problem with Time Windows (OPTW) in which a given positive profit and time interval are associated with each location. This paper approaches OPTW suggesting a comparison between two different methods to solve it: Biased Random Key Genetic Algorithm (BRKGA) and MultiStart. Both techniques use a local search with four steps, based on the last solutions proposed to this problem. Computational experiments were executed on traditional instances. The tests showed that the BRKGA method achieved larger or same results than MultiStart for all the 76 instances evaluated. Furthermore, were found 21 new best solutions for OPTW instances, in comparison with the current literature.*

Keywords: *BRKGA, MultiStart, OPTW, TTDP.*

Resumo. *O Problema de Planejamento de Rotas Turísticas (TTDP) é um problema computacional que envolve o planejamento de rotas entre pontos de interesse de uma região turística. Ele pode ser modelado como um problema de roteamento bem conhecido, chamado Problema de Orientação com Janelas de Tempo (OPTW), no qual um score e um intervalo de tempo são associados a cada localidade. Este artigo aborda o OPTW ao propor uma comparação entre dois diferentes métodos para solucioná-lo: Biased Random Key Genetic Algorithm (BRKGA) e MultiStart. Ambas técnicas utilizaram a mesma busca local com quatro etapas, baseada nas últimas soluções propostas para este problema. Foram executados experimentos computacionais com as instâncias tradicionais. Os testes mostraram que o método BRKGA atingiu resultados iguais ou melhores que MultiStart em todas as 76 instâncias avaliadas. Além disso, foram encontradas 21 novas melhores soluções para OPTW, em comparação com a literatura atual.*

Palavras chave: *BRKGA, MultiStart, OPTW, TTDP.*

1 INTRODUÇÃO

O planejamento de rotas turísticas envolve uma pesquisa substancial para selecionar pontos de interesse (do inglês: Points of Interests - POIs) para serem visitados, levando em consideração o tempo despendido em cada atração e o tempo necessário para se movimentar entre um local e outro. Este tema é abordado por meio do Problema do Planejamento de Rotas Turísticas (do inglês: *Touristic Trip Design Problem* - TTDP), que se ocupa em propor roteiros de pontos de visitação adequados ao tempo disponível pelo viajante em seu passeio.

O TTDP é tratado como uma extensão do Problema de Orientação (do inglês: *Orienteering Problem* - OP) (Vansteenwegen, 2007), que, por sua vez, foi introduzido por Tsiligirides (1984), a partir de uma das modalidades de competição do esporte homônimo. Nela, o atleta deve percorrer um terreno marcando presença em pontos de controle indicados em um mapa, dentro do tempo estipulado pela organização. Os pontos de controle possuem pesos diferenciados, cuja soma resulta na pontuação (*score*) do competidor. Cabe a ele decidir quantos e quais pontos de controle serão incluídos em seu trajeto.

De acordo com Vansteenwegen et al (2011), o Problema de Orientação consiste na determinação do menor caminho hamiltoniano entre vértices ponderados selecionados. No OP, não é necessário cobrir todos os vértices e aqueles selecionados podem ser percorridos apenas uma vez.

O Problema de Orientação é conhecido também por *Selective Travelling Salesperson Problem*, *Maximum Collection Problem* e *Bank Robber Problem*. Suas aplicações envolvem casos de decisão de atendimento de clientes em diferentes cidades (Tsiligirides, 1984), de distribuição de combustível (Golden et al, 1987) e a versão mais simplificada do TTDP (Vansteenwegen; Van Oudheusden, 2007).

Neste trabalho são propostas duas soluções para OPTW: uma baseada na metaheurística *Biased Random Key Genetic Algorithm* (BRKGA) e outra no algoritmo MultiStart. Na seção 2, apresenta-se o problema em discussão e uma revisão da literatura. A seção 3 descreve o algoritmo BRKGA, ao passo que a seção 4 exhibe os resultados dos experimentos realizados e a seção 5 apresenta as conclusões.

2 PROBLEMA DE ORIENTAÇÃO COM JANELAS DE TEMPO

OP é definido por Vansteenwegen et al. (2011) a partir de um conjunto com uma quantidade N de vértices i , cada um com uma pontuação não negativa S_i . Assim, $i = 1$ é o ponto inicial, $i = N$ é o ponto final da rota e a pontuação de ambos é nula ($S_1 = S_N = 0$). O tempo de deslocamento entre os vértices i e j é conhecido por t_{ij} e o tempo disponível para realizar o percurso entre os vértices selecionados é limitado por T_{max} . Há ainda as variáveis binárias x_{ij} , que recebem valor igual a 1 se, partindo do vértice selecionado i , o ponto j for o próximo a ser visitado; do contrário, recebem valor igual a 0. u_i representa a posição do vértice selecionado i na rota da solução. Considera-se os deslocamentos de forma simétrica entre os vértices ($t_{ij} = t_{ji}$). O objetivo do OP é determinar um trajeto, limitado por T_{max} , que visite alguns vértices, de forma a maximizar a pontuação final da solução.

Quando se acrescenta ao OP restrições temporais para o funcionamento de cada vértice, isto é, existe um intervalo de tempo para o qual cada ponto pode ser visitado, o problema recebe o nome de Orientação com Janelas de Tempo (do inglês: *Orienteering Problem with Time Windows* – OPTW) (Kantor e Rosenwein, 1992). Deste modo, é necessário associar a cada vértice uma janela de tempo $[O_i, C_i]$, de forma que uma visita possa ser iniciada apenas durante este intervalo. Além disso, inclui-se também ao problema as variáveis y_i , que recebem valor 1, se o vértice i faz parte da rota selecionada, ou 0, caso contrário; as variáveis v_i , que indicam o momento de início da visita ao vértice i ; e uma constante grande M . Assim, a formulação matemática do OPTW é dada por (Vansteenwegen et al., 2011):

$$\text{Max} \sum_{i=2}^{N-1} \sum_{j=2}^N S_i x_{ij}, \quad (1)$$

$$\sum_{j=2}^N x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \quad (2)$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^N x_{kj} \leq 1; \quad \forall k = 2, \dots, N-1, \quad (3)$$

$$v_i + t_{ij} - v_j \leq M(1 - x_{ij}); \quad \forall i, j = 1, \dots, N, \quad (4)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N t_{ij} x_{ij} \leq T_{max}, \quad (5)$$

$$O_i \leq v_i; \quad \forall i = 1, \dots, N, \quad (6)$$

$$v_i \leq C_i; \quad \forall i = 1, \dots, N, \quad (7)$$

$$x_{ij} \in \{0,1\}; \quad \forall i, j = 1, \dots, N \quad (8)$$

Na Eq. (1), a função objetivo maximiza a pontuação coletada. As restrições na Eq. (2) especificam que a rota da solução inicia e termina nos vértices 1 e N , respectivamente. Equação (3) determina a conectividade do trajeto e cria restrições para que cada vértice seja percorrido no máximo uma vez. Na Eq. (4) garante-se a sequência temporal do percurso e na Eq. (5) limita-se o tempo gasto para percorrê-lo. As restrições presentes nas Eq. (6) e (7) determinam a janela temporal de visitação de cada vértice.

Na realidade, a maioria das aplicações do OP envolve janelas de tempo e, assim, tornam-se aplicações para o OPTW. No TTDP, por exemplo, é necessário considerar o horário de funcionamento de cada local a ser visitado. Da mesma forma, no problema de distribuição de combustível é preciso que o estabelecimento do cliente esteja em atividade no momento da entrega.

Além disso, existe, ainda, uma variação do OPTW em que são permitidas mais de uma rota para percorrer os vértices do problema, sem repeti-los; chamada *Team Orienteering Problem with Time Windows* (TOPTW). O OPTW se equivale ao TOPTW quando o número de rotas é 1 (Gunawan et al., 2016).

2.1 Revisão de Literatura

Golden et al. (1987) atestaram que o OP é um problema NP-Completo, isto é, não existe um algoritmo ótimo capaz de resolvê-lo em tempo polinomial. Deste modo, soluções exatas para o OP não são viáveis, o que implica no emprego de heurísticas e meta-heurísticas para obtenção de soluções aceitáveis em um curto tempo de processamento. Sobre isto, Gendreau et al. (1998) afirmaram que, apesar da aparente simplicidade do OP, é difícil elaborar heurísticas para este problema, uma vez que as pontuações dos vértices são independentes das distâncias entre eles e, em geral, obtêm-se estratégias de solução adequadas apenas a uma dessas variáveis. Analogamente, o OPTW também é um problema NP-Completo. Porém, a existência das restrições temporais exige a adaptação de soluções desenvolvidas para o OP (Vansteenwegen et al., 2011).

Nos últimos anos, diversos trabalhos têm abordado o OP e suas variações. A investigação elaborada por Gunawan et al. (2016) se dedica a referenciar os artigos publicados nos últimos 5 anos relacionados às variações desse problema, incluindo a OPTW e TOPTW.

Vansteenwegen et al. (2009) introduziram um algoritmo de *Iterated Local Search* (ILS) para resolver o TOPTW, abordando duas operações (INSERT e SHAKE).

Um método híbrido entre *Greed Randomized Adaptive Search Procedure* (GRASP) e *Evolutionary Local Search* (ELS) foi publicado por Labadie et al. (2011). O algoritmo utiliza diferentes heurísticas construtivas baseadas em GRASP para construir as soluções iniciais que alimentarão o algoritmo ELS.

Gambardella et al. (2012) desenvolveram uma metaheurística, *Enhanced Ant Colony System*, baseada no método de colônia de formigas, introduzindo uma seleção de soluções a serem submetidas à busca local.

Lin e Yu (2012) construíram duas variações da metaheurística *Simulated Annealing* (SA): uma versão, FSA, capaz de produzir resultados de forma rápida para aplicações; e outra, SSA, voltada para a obtenção de soluções de melhor qualidade e maior tempo computacional. Para solucionar OPTW e TOPTW, Labadie et al. (2012) propuseram um algoritmo de *Granular Variable Neighborhood Search* (GVNS).

Seguindo a corrente de utilização de algoritmos híbridos, Souffriau et al. (2013) apresentaram o GRILS: uma composição entre os métodos GRASP e ILS. Apesar de ter sido desenvolvido para uma variante do TOPTW, também foi executado para casos do problema originário.

Hu e Lim (2014) criaram o método híbrido I3CH, composto por três componentes: busca local, SA e *Route Recombination* (RR). Executados a cada iteração, o método obteve 35 novas melhores soluções.

Cura (2014) propôs a utilização da metaheurística *Artificial Bee Colony* (ABC) como solução do TOPTW. Um método baseado em SA e em comportamentos de abelhas foi incorporado para melhorar a qualidade dos resultados.

Outro algoritmo ILS foi elaborado por Gunawan et al. (2015a). Nele, a solução inicial é elaborada por meio de uma heurística gulosa que é submetida ao método de busca local com etapas SWAP, 2-OPT, INSERT e REPLACE. Um mecanismo de equilíbrio entre diversificação ou melhoramento das soluções é adotado. Uma nova versão do algoritmo, com mais procedimentos na fase de busca local é apresentada em Gunawan et al. (2015c). Por fim, foi desenvolvida uma hibridização entre o método ILS e o SA, denominada SAILS, para tornar o algoritmo capaz de escapar de ótimos locais (Gunawan, 2015b).

2.1.1 Busca Local

A solução desenvolvida adota um algoritmo de busca local baseado no utilizado por Gunawan (2015c), para TOPTW. Neste, realizam-se seis operações consecutivas para aprimorar as soluções obtidas em cada iteração. O procedimento foi adaptado para a condição do OPTW, restando apenas quatro operações para a busca local, como descrito na Tabela 1.

Tabela 1. Operações da busca local

Operação	Descrição
SWAP	Permuta dois vértices de uma rota
2-OPT	Inverte a ordem de uma sequência de nós de uma rota
INSERT	Insere vértices em uma rota
REPLACE	Substitui um vértice de uma rota por outro ainda não percorrido

Inicialmente, a operação SWAP é executada, trocando a posição de dois vértices de uma rota, em todas as possibilidades de combinações de permutação. Se a nova configuração do percurso for válida e se o somatório do tempo de espera de abertura dos vértices da nova possibilidade for maior que o da rota original, então ocorre a substituição da solução.

Na sequência, ocorre a operação 2-OPT. Dois vértices de uma rota são selecionados e inverte-se o sentido do percurso entre eles. Se a nova rota for válida e se o novo tempo de espera for maior que o original, então ocorrerá a substituição.

Em seguida, acontece a operação INSERT, responsável por inserir vértices ainda não utilizados na solução original. A seleção e ordenamento de quais vértices serão submetidos ao processo de inserção é realizada por meio do Algoritmo 1. Nele, N' refere-se aos vértices não utilizados pela rota m , e $P(m)$ corresponde às posições dos vértices do percurso da rota m . Além disso, f é um parâmetro configurável.

Algoritmo 1. UpdateF(N', m)

```

1 -  $F \leftarrow 0$ 
2 - Para todo  $n \in N'$  faça
3 -     Para todo  $p \in P(m)$  faça
4 -         Se inserir vértice  $n$  na posição  $p$  da rota  $m$  for possível então
5 -             Calcular  $ratio_{n,p}$ 
6 -              $F \leftarrow F \cup \langle n, p \rangle$ 
7 -         FimSe
8 -     FimPara
9 - FimPara
10 - Ordenar todos os elementos de  $F$  decrescentemente com base em  $ratio_{n,p}$ 
11 - Selecionar os  $f$  melhores elementos de  $F$  e remover o resto
12 - Retornar  $F$ 

```

Sendo que $ratio_{n,p}$ (linha 5) corresponde à Eq. (9), em que u_n é a pontuação obtida pela nova rota alterada e $\Delta_{n,p}$ representa a diferença entre o tempo total gasto antes e depois da inserção do novo vértice.

$$ratio_{n,p} = \frac{u_n^2}{\Delta_{n,p}} \quad (9)$$

Após a execução do Algoritmo 1, ainda no escopo da operação INSERT, executa-se o algoritmo 2, que realiza a seleção do vértice candidato a ser inserido nas posições da rota, em função do peso de seu $ratio_{n,p}$. O procedimento se repete até que $F = 0$.

Algoritmo 2. Select(F)

```

1 -  $SumRatio \leftarrow 0$ 
2 - Para todo  $\langle n, p \rangle \in F$  faça
3 -      $SumRatio \leftarrow SumRatio + ratio_{n,p}$ 
4 - FimPara
5 - Para todo  $\langle n, p \rangle \in F$  faça
6 -      $prob_{n,p} \leftarrow ratio_{n,p} / SumRatio$ 
7 - FimPara
8 -  $U \leftarrow rand(0,1)$ 
9 -  $AccumProb \leftarrow 0$ 
10 - Para todo  $\langle n, p \rangle \in F$  faça
11 -      $AccumProb \leftarrow AccumProb + prob_{n,p}$ 
12 -     Se  $(U \leq AccumProb)$  então
13 -          $\langle n^*, p^* \rangle \leftarrow \langle n, p \rangle$ 
14 -     Interrompa
15 -     FimSe
16 - FimPara
17 - Retorne  $\langle n^*, p^* \rangle$ 

```

Por fim, a última operação, REPLACE, busca substituir um vértice da rota em questão por outro ainda não utilizado. Se a mudança gerar uma rota viável e melhorar a pontuação acumulada da solução, então se aplica a mudança. Esta etapa termina quando todos os vértices não utilizados forem verificados.

3 ESTRATÉGIA DE SOLUÇÃO: HEURÍSTICA BRKGA-OPTW

Motivados pelo sucesso da utilização de Algoritmos Genéticos Baseados em Chaves Aleatórias para a resolução de problemas de otimização combinatória em Arulselvan et al. (2007), Buriol et al. (2005), Buriol et al. (2007), Gonçalves et al. (2005), Gonçalves et al. (2004), Gonçalves et al. (2009), Goulart et al. (2011), Goulart et al. (2011b), Malve e Uzsoy (2007), Noronha et al. (2010), Reis et al. (2011), Samanlioglu et al. (2008), Snyder et al. (2006), optou-se por adaptar um modelo BRKGA, resultando na heurística BRKGA-OPTW.

BRKGA-OPTW utiliza a heurística construtiva e busca local proposta em Gunawan et al. (2015c) para a criação de uma população de cromossomos, compostos por *arrays* de POIs a serem visitados. Cada um destes recebe um valor aleatório entre 0 e 1, utilizado para ordená-los de forma crescente. Esta sequência de POIs resultante é enviada como parâmetro para a heurística gulosa de Gunawan et al. (2015c), submetendo-a a uma busca local. A solução retornada pela heurística passa a ser, então, o valor de adaptabilidade do cromossomo.

A população inicial de cromossomos é gerada de maneira aleatória. A partir disso, seguindo a lógica de um algoritmo genético, a cada geração, a população é dividida em dois conjuntos, denominados *TOP* e *REST*. A parcela *TOP* contém os indivíduos com as melhores soluções e *REST* agrupa os remanescentes.

Para o povoamento de cada geração, é formada uma nova população. Os indivíduos *TOP* são copiados, sem alterações. No entanto, o conjunto *REST* é substituído por dois outros conjuntos: *MID* e *BOT*. *BOT* sobrepõe os piores indivíduos de *REST*, sendo composto por novos cromossomos produzidos de forma aleatória, o que possibilita escapar de mínimos locais. O restante, ou *MID*, é preenchido por outros indivíduos elaborados pela aplicação de operadores de cruzamento e mutação.

O operador de cruzamento *Parametrized Uniform Crossover*, proposto por Spears e de Jong (1991), foi utilizado em BRKGA-OPTW. O cruzamento é realizado entre um indivíduo da parcela *TOP* da população e outro da parcela *REST*. Para cada chave do *array* do cromossomo em construção, a probabilidade de herdá-la da parcela *TOP* é de 70%, ao passo que para *REST* é de 30%.

Assim, o tamanho total da população inicial é $TOP+REST$ e a quantidade de indivíduos criados a cada geração através de cruzamento é $MID = REST - BOT$ cromossomos filhos.

Os tamanhos das parcelas *TOP*, *REST* e *BOT* são informados através de parâmetros do algoritmo. O pseudocódigo da heurística BRKGA-OPTW é exibido no Algoritmo 3.

Algoritmo 3. BRKGA – OPTW($POIs, p_{rec}$)

```

1 - Para cada vértice do conjunto de  $POIs$  faça
2 - Gerar vetor com chaves aleatórias;
3 - Avaliar vetor de acordo com a função de custo;
4-FimPara
5 - Enquanto critério de parada não for satisfeito faça
6- Ordenar a população pelo valor retornado;
7 - Dividi-la em  $TOP$ ,  $MID$  e  $BOT$ ;
8 - Copiar a parcela  $TOP$  para a população da próxima geração;
9 -Para cada elemento da parcela  $MID$  faça
10 -Para cada chave do novo cromossomo faça
11 - Selecionar aleatoriamente um elemento  $p_1$  da parcela  $TOP$ ;
12 - Selecionar aleatoriamente um elemento  $p_2$  da parcela  $MID \cup BOT$ ;
13 -  $p_3 \leftarrow Crossover(p_1, p_2, p_{rec})$ ;
14 - Adicionar  $p_3$  à parcela  $MID$  da nova geração;
14-FimPara
15-FimPara
16 - Gerar aleatoriamente a parcela  $BOT$  da nova geração;
17 - Avaliar cada elemento da população;
18-FimEnquanto

```

Inicialmente, a heurística BRKGA-OPTW recebe como parâmetros os pontos de interesse da instância ($POIs$), e o percentual de cruzamento selecionado (p_{rec}). O laço entre as linhas 1 à 4 é executado até que sejam avaliados cada um dos pontos de interesse do conjunto de $POIs$. Na linha 2, os vetores de chaves são inicializados aleatoriamente. Na linha 3, avalia-se a função de custo para cada vetor. O laço entre as linhas 5 e 18 é executado até que o critério de parada seja alcançado. Na linha 6, a população é ordenada de acordo com a função de custo. Na linha 8, a parcela TOP da população é integralmente copiada para a próxima geração. Entre as linhas 9 e 15 é realizado os processos de seleção e cruzamento, gerando os cromossomos da porção MID da nova população, em que deve-se ressaltar a linha 13, onde os cromossomos p_1 e p_2 produzem o indivíduo p_3 através da probabilidade p_{rec} de cruzamento. Na linha 16 é gerado aleatoriamente o novo conjunto BOT . Na linha 17, a nova população é avaliada.

4 EXPERIMENTOS

A estratégia proposta foi comparada a um algoritmo MultiStart, no qual obtém-se, a cada iteração, uma sequência de chaves gerada aleatoriamente e que é submetida à mesma heurística de construção e busca local de BRKGA-OPTW, descrita anteriormente.

Os experimentos realizados foram codificados em linguagem de programação C++ e o compilador GNU GCC versão 4.4.3, utilizando a busca local. Para a implementação de BRKGA-OPTW, utilizou-se a biblioteca *brkgaAPI*, desenvolvida por Toso e Resende (2015). Foram adotadas 76 instâncias clássicas para OPTW, obtidas pela biblioteca *The Orienteering Problem: Test Instances*, disponível em <http://www.mech.kuleuven.be/en/cib/op>. Foram executadas 5 repetições para cada

instância, em um computador com processador AMD A10 PRO-7800B, de 3,50 GHz e 8,00 GB de memória RAM.

Seguindo as recomendações da API, os tamanhos dos conjuntos TOP, REST e BOT foram configurados como $0,20 \cdot |X|$; $0,80 \cdot |X|$; e $0,1 \cdot |X|$, respectivamente, sendo $|X|$ a quantidade de indivíduos da população, igual a 1000 cromossomos. O critério para parada de execução da heurística BRKGA-OPTW foi configurado para 1000 gerações. Já o MultiStart foi configurado para ser finalizado após obter $1000 \cdot |X| = 1000000$ soluções. O parâmetro F da busca local foi definido como 5, como indicado por Gunawan (2015c).

Os resultados numéricos dos experimentos estão listados na Tabela 2.

Tabela 2. Resultados detalhados obtidos para BRKGA-OPTW

Instância	BRKGA			MultiStart			Gap (%)
	Melhor	Média	Desvio Padrão	Melhor	Média	Desvio Padrão	
c101	320	320,0	0,0	270	264,0	4,9	21,21
c102	360	360,0	0,0	310	310,0	0,0	16,13
c103	380	380,0	0,0	380	372,0	4,0	2,15
c104	390	390,0	0,0	390	390,0	0,0	0,00
c105	340	340,0	0,0	290	286,0	4,9	18,88
c106	340	340,0	0,0	300	298,0	4,0	14,09
c107	370	368,0	4,0	320	316,0	4,9	16,46
c108	380	376,0	4,9	320	320,0	0,0	17,50
c109	380	380,0	0,0	370	356,0	8,0	6,74
r101	205	205,0	0,0	171	166,4	3,0	23,20
r102	289	289,0	0,0	246	239,6	4,8	20,62
r103	299	294,2	2,4	245	242,6	2,1	21,27
r104	310	304,8	2,8	256	247,2	5,1	23,30
r105	255	255,0	0,0	204	196,4	4,8	29,84
r106	293	292,4	0,8	258	251,2	6,1	16,40
r107	304	300,6	3,1	253	247,0	3,6	21,70
r108	309	308,0	1,1	253	245,4	4,1	25,51
r109	287	285,4	2,1	231	217,4	7,7	31,28
r110	289	285,6	1,7	227	216,4	7,4	31,98
r111	307	300,8	3,1	255	246,6	5,9	21,98
r112	302	299,0	2,7	246	239,8	4,6	24,69
rc101	219	219,0	0,0	209	203,0	4,1	7,88
rc102	266	263,0	3,7	222	215,2	5,0	22,21
rc103	268	262,0	4,8	220	213,0	4,1	23,00
rc104	301	294,6	7,8	234	230,6	3,1	27,75
rc105	244	244,0	0,0	205	203,4	2,7	19,96
rc106	253	251,6	2,8	213	212,0	1,3	18,68
rc107	278	277,6	0,5	240	232,0	4,6	19,66
rc108	298	283,2	12,4	246	235,2	7,3	20,41
pr01	312	307,8	2,4	211	207,2	3,5	48,55
pr02	413	402,6	7,2	272	257,0	9,4	56,65
pr03	396	382,0	8,5	271	245,8	14,4	55,41
pr04	484	471,6	9,3	288	272,6	8,0	73,00
pr05	546	525,2	17,4	279	268,4	8,5	95,68
pr06	526	504,2	12,9	275	269,8	4,4	86,88
pr07	301	299,4	3,2	219	213,8	2,7	40,04
pr08	474	454,4	14,6	274	261,0	7,1	74,10
pr09	475	461,2	12,4	242	238,8	3,1	93,13
pr10	520	484,8	18,9	284	269,2	7,7	80,09

Instância	BRKGA			MultiStart			Gap (%)
	Melhor	Média	Desvio Padrão	Melhor	Média	Desvio Padrão	
c201	860	844,0	13,6	410	404,0	8,0	108,91
c202	910	884,0	21,5	510	498,0	7,5	77,51
c203	940	920,0	22,8	700	682,0	9,8	34,90
c204	900	888,0	7,5	790	790,0	0,0	12,41
c205	880	844,0	25,8	480	472,0	7,5	78,81
c206	920	908,0	14,7	530	502,0	16,0	80,88
c207	920	904,0	10,2	560	550,0	8,9	64,36
c208	920	920,0	0,0	570	548,0	13,3	67,88
r201	785	770,2	12,0	405	382,4	14,1	101,41
r202	880	866,4	12,0	529	520,8	5,3	66,36
r203	898	885,8	9,6	605	595,8	7,9	48,67
r204	984	949,0	14,2	652	651,0	1,0	45,78
r205	889	871,6	13,7	549	502,6	24,2	73,42
r206	961	924,0	24,1	612	599,6	8,5	54,10
r207	997	960,2	26,6	644	628,2	8,6	52,85
r208	945	942,0	2,9	678	666,5	11,5	41,34
r209	900	878,6	17,2	532	527,6	4,2	66,53
r210	932	912,2	10,6	566	560,8	4,4	62,66
r211	925	910,4	7,6	596	588,6	6,2	54,67
rc201	795	785,0	8,4	432	413,0	11,6	90,07
rc202	900	867,8	18,6	501	488,8	7,1	77,54
rc203	916	891,4	16,2	565	550,4	8,8	61,95
rc204	1010	987,8	17,7	627	615,8	9,2	60,41
rc205	828	798,8	26,7	464	459,8	8,4	73,73
rc206	887	847,0	25,1	502	495,0	5,0	71,11
rc207	914	891,4	16,2	542	529,4	7,0	68,38
rc208	935	910,4	13,8	615	592,4	12,8	53,68
pr11	342	336,6	3,7	255	246,0	5,1	36,83
pr12	436	434,8	1,9	283	280,4	1,7	55,06
pr13	450	434,2	12,7	276	271,0	3,0	60,22
pr14	544	520,2	15,0	285	277,8	4,5	87,26
pr15	614	594,6	12,2	320	310,0	8,4	91,81
pr16	583	560,3	19,7	315	307,5	7,5	82,21
pr17	364	357,0	5,2	280	255,6	12,4	39,67
pr18	525	503,8	17,1	287	282,4	3,3	78,40
pr19	487	464,2	16,2	296	285,0	6,1	62,88
pr20	574	546,0	25,8	313	299,8	11,1	82,12

A primeira coluna informa o nome da instância. As próximas três colunas apresentam o *score*, o *score* médio e o desvio padrão obtidos utilizando a heurística BRKGA-OPTW. As três colunas seguintes apresentam o *score*, o *score* médio e o desvio padrão obtidos utilizando o método MultiStart. A coluna “Gap” refere-se à melhora relativa entre os métodos, sendo a razão entre a diferença dos valores médios de BRKGA-OPTW e MultiStart, pelo valor médio de MultiStart.

Os *scores* médios obtidos por BRKGA-OPTW foram até 108,91% superiores aos apresentados por MultiStart, na instância c201, e 48,62% em média, para todo o conjunto de instâncias considerado. Deve-se ressaltar ainda que, em todas as instâncias, a média dos *scores* obtidos por BRKGA-OPTW é sempre maior ou igual aos *scores* obtidos por MultiStart.

Comparando-se os melhores resultados obtidos por BRKGA-OPTW com os melhores *scores* da literatura atual (do inglês, *Best Known* (BK)), disponíveis em Gunawan et al. (2015d), foram encontradas 21 novas melhores soluções, como descrito na Tabela 3.

Tabela 3: Novas melhores soluções obtidas por BRKGA-OPTW

Instância	BK Anterior	Novo BK	Instância	BK Anterior	Novo BK
c108	370	380	r105	247	255
pr01	308	312	r107	299	304
pr02	404	413	r108	308	309
pr03	394	396	r109	277	287
pr07	298	301	r110	284	289
pr08	463	474	r111	297	307
pr17	362	364	r112	298	302
r101	198	205	rc103	266	268
r102	286	289	rc106	252	253
r103	293	299	rc107	277	278
r104	303	310			

Além disso, em 13 instâncias BRKGA-OPTW também atingiu o mesmo valor de *score* dos BKs atuais.

5 CONCLUSÕES

Este trabalho abordou o *Orienteering Problem with Time Windows*, propondo uma nova heurística BRGA-OPTW e comparando-a com o método MultiStart, ambos com uma busca local previamente existente, adaptada de Gunawan et al. (2015c).

Os experimentos computacionais mostraram a eficácia do método proposto em todas as instâncias avaliadas, igualando ou superando os resultados obtidos com MultiStart. Além disso, a heurística BRKGA-OPTW obteve 21 novas melhores soluções, em comparação com a literatura recente.

REFERÊNCIAS

- Arulseivan, A., Commander, C., & Pardalos, P., 2007. A random keys based genetic algorithm for the target visitation problem. Pardalos, P., Murphey, R., Grundel, D. & Hirsch, M., eds, *Advances in Cooperative Control and Optimization*, vol. 369 of Lecture Notes in Control and Information Sciences, pp. 389–397. Springer.
- Buriol, L. S., Resende, M. G. C., Ribeiro, C. C., & Thorup, M., 2005. A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. *Networks*, 46:36–56.
- Buriol, L. S., Resende, M. G. C., & Thorup, M., 2007. Survivable IP network design with OSPF routing. *Networks*, 49:51–64.
- Cura, T., 2014. An artificial bee colony algorithm approach for the team orienteering problem with time windows. *Computers & Industrial Engineering*, vol. 74, pp. 270–290.
- Gambardella, L. M., Montemanni, R., & Weyland, D., 2012. Coupling ant colony systems with strong local searches. *European Journal of Operational Research*, vol. 220, n. 3, pp. 831–843.
- Gendreau, M., Laporte, G., & Semet, F., 1998. A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, vol. 106, n. 2-3, pp. 539–545.
- Golden, B. L., Levy, L., & Vohra, R., 1987. The orienteering problem. *Naval Research Logistics*, vol. 34, n. 3, pp. 307–318.
- Gonçalves, J. F., Mendes, J. J. M., & Resende, M. G. C., 2005. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, vol. 167, n. 1, pp. 77–95.
- Gonçalves, J. F. & Resende, M. G. C., 2004. An evolutionary algorithm for manufacturing cell formation. *Computers and Industrial Engineering*, vol. 47, pp. 247–273.
- Gonçalves, J. F., & Resende, M. G. C., 2009. Biased random-key genetic algorithms for combinatorial optimization. To appear in *J. of Heuristics*, v. 17, n. 5, pp. 487–525.
- Goulart, N., Noronha, T. F., de Souza, S. R., & Dias, L. G. S., 2011. Algoritmo Genético para o Problema de Instalação de Fibras em Redes Óticas. XLIII Simpósio Brasileiro de Pesquisa Operacional, Ubatuba.
- Goulart, N., Noronha, T. F., de Souza, S. R., & Dias, L. G. S., 2011. Biased Random-key Genetic Algorithm for Fiber Installation in Optical Network Optimization. 2011 IEEE *Evolutionary Computation*, New Orleans, LA, 2011, pp. 2267–2271.
- Gunawan, A., Lau, H. C., & Lu, K., 2015a. An iterated local search algorithm for solving the orienteering problem with time windows. In Ochoa, G., Chicano, F., eds, *Evolutionary Computation in Combinatorial Optimization*, vol. 9026 of Lecture Notes in Computer Science (Springer), pp. 61–73.
- Gunawan, A., Lau, H. C., & Lu, K., 2015b. SAILS: hybrid algorithm for the team orienteering problem with time windows. In *Proceedings of the 7th Multidisciplinary International Scheduling Conference (MISTA 2015)*. Prague, Czech Republic, pp. 276–295.
- Gunawan, A., Lau, H. C., & Lu, K., 2015c. Well-tuned ILS for extended team orienteering problem with time windows. *LARC technical report series*, Singapore Management University.

- Gunawan, A., Lau, H. C., & Lu, K., 2015d. The latest best known solutions for the Team Orienteering Problem with Time Windows (TOPTW) benchmark instances. Disponível em <http://research.larc.smu.edu.sg/downloads/OPLib/Publications/SupplementTOPTW.pdf>. Acesso em 15 de junho de 2016.
- Gunawan, A., Lau, H. C., & Vansteenwegen, P., 2016. Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, vol. 255, n.2, pp.315–332.
- Hu, Q., & Lim, A., 2014. An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, vol. 232, n. 2, pp. 276–286.
- Kantor, M. G., & Rosenwein, M. B., 1992. The Orienteering Problem with Time Windows. *The Journal of the Operational Research Society*, vol. 43, n. 6, pp. 629–635.
- Labadie, N., Mansini, R., Melechovský, J., WolflerCalvo, R., 2011. *Hybridized evolutionary local search algorithm for the team orienteering problem with time windows*. *Journal of Heuristics*, vol. 17, n. 6, pp. 729–753.
- Labadie, N., Mansini, R., Melechovský, J., & WolflerCalvo, R., 2012. The team orienteering problem with time windows: an LP-based granular variable neighborhood search. *European Journal of Operational Research*, vol. 220, n. 1, pp. 15–27.
- Lin, S. W., Yu, & V. F., 2012. A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, vol. 217, n. 1, pp. 94–107.
- Malve, S., & Uzsoy, R., 2007. A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families. *Computers & Operations Research*, v. 34, p. 3016–3028.
- Noronha, T. F., Resende, M. G. C., & Ribeiro, C. C., 2010. A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, vol. 50, n. 3, pp. 503-518.
- Reis, R., Ritt, M., Buriol, L. S., & Resende, M. G. C., 2011. A biased random-key genetic algorithm for ospf and deft routing to minimize network congestion. *International Transactions in Operational Research*, vol. 18, pp. 401–423.
- Samanlioglu, F., Ferrell, W. G. Jr., & Kurz, M. E., 2008. A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem. *Computers & Industrial Engineering*, v. 55, n. 2, p. 439–449.
- Snydera, L. V., & Daskin, M. S., 2006. A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research*, vol. 174, pp. 38–53.
- Souffriau, W., Vansteenwegen, P., VandenBerghe, G., & Van Oudheusden, D., 2013. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science*, vol. 47, n. 1, pp. 53–63.
- Spears, W., de Jong, K., 1991. On the virtues of parameterized uniform crossover. *Proceedings of the Fourth International Conference on Genetic Algorithms*, p. 230–236, San Mateo.
- The Orienteering Problem: Test Instances*, 2011. Disponível em <http://www.mech.kuleuven.be/en/cib/op>. Acessado em 15 de junho de 2016.

Toso, Rodrigo F., Resende, Mauricio G. C., 2015. A C++ Application Programming Interface for Biased Random-Key Genetic Algorithms. *Optimization Methods and Software*, vol. 30, n. 1, pp. 81-93.

Tsiligirides, T., 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, vol. 35, n. 9, pp. 797–809.

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D., 2011. The Orienteering Problem: a Survey. *European Journal of Operational Research*, vol. 209, n. 1, pp. 1–10.

Vansteenwegen, P., Souffriau, W., VandenBerghe, G., & Van Oudheusden, D., 2009. Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, vol. 36, n. 12, pp. 3281–3290.

Vansteenwegen, P., & Van Oudheusden, D., 2007. The mobile tourist guide: An OR opportunity. *Operational Research Insight*, vol. 20, n. 3, pp. 21–27.