Algoritmo Genético Aplicado à Otimização do Planejamento de Torneios Esportivos

Paulo Henrique dos Santos
Departamento de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto, Minas Gerais, 35400-000, Brasil.
paulosantosphs@hotmail.com

Marco Antonio Moreira de Carvalho
Departamento de Ciência da Computação, Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto, Minas Gerais, 35400-000, Brasil.
mamc@ufop.edu.br

RESUMO

Relacionado aos problemas de planejamento de torneios esportivos, o *Mirrored Traveling Tournament Problem* (mTTP) é um problema de otimização $\mathcal{NP}$-Difícil que visa a produção de calendários de competições esportivas com o objetivo de minimizar a distância total viajada por todas as equipes do campeonato. Neste artigo é apresentada uma abordagem que aplica a metaheurística o Algoritmo Genético de Chaves Aleatórias Viciadas em conjunto com o método do polígono para a solução do mTTP. Os experimentos computacionais envolveram os dois principais conjuntos de instâncias reportados na literatura e os resultados gerados demonstram a competitividade do método proposto.

PALAVRAS CHAVE. Escalonamento de Torneios, Traveling Tournament Problem, BRKGA.

Área Principal: L&T, MH, OC

ABSTRACT

Related to sports scheduling problems, the Mirrored Traveling Tournament Problem (mTTP) is an optimization problem classified as $\mathcal{NP}$-Hard which aims to produce the schedule of sports tournaments with the objective of minimizing the total distance traveled by all the teams of the tournament. In this paper we present an approach that applies the metaheuristic Biased Random-Key Genetic Algorithm associated with the Polygon Method in ordeer to solve the mTTP. The computational experiments involved the two main sets of instances from the literature and the results generated demonstrates the competitiveness of the proposed method.

KEYWORDS. Tournament Scheduling, Traveling Tournament Problem, BRKGA.

Main Area: L&T, MH, OC

https://proceedings.science/p/85467?lang=pt-br

1. Introdução

Vários esportes possuem campeonatos longos, divididos em temporadas de aproximadamente um ano e que são organizados pelas federações ou ligas. As temporadas ainda podem ser subdivididas em *turnos* e *rodadas*. Estes campeonatos são tipificados de acordo com a estrutura da *tabela* de confrontos, ou *calendário*. Por exemplo, caso o campeonato possua uma estrutura em que todas as equipes confrontam todas as outras um número fixo de vezes, caracteriza-se o tipo *Round Robin*. Caso este número de encontros entre as equipes seja exatamente um, o campeonato é do tipo *Single Round Robin* (SRR). Caso este número seja exatamente dois, o campeonato é do tipo *Double Round Robin* (DRR). Como ilustração, cita-se o Campeonato Brasileiro de Futebol, em que as equipes se confrontam, em dois turnos.

O planejamento destes campeonatos, no entanto, vai muito além destas estruturas básicas. Existem diferentes variáveis que afetam diretamente os interesses das equipes, organizadores, patrocinadores e torcedores. Por exemplo, o direito de sediar um confronto é extremamente importante. O padrão de alternância de confrontos em que que cada equipe joga dentro ou fora de casa (*Home-Away Pattern*, HAP) por exemplo, influencia diretamente a logística e arrecadação de bilheterias, além do benefício de jogar junto à torcida local.

Desta maneira, os organizadores destes se preocupam em produzir um calendário que atenda também diversos objetivos práticos e logísticos, eventualmente conflitantes. Um objetivo comum é a produção de um calendário justo em que todas as equipes possuam o mesmo número de confrontos como mandante (i.e., "em casa") ou o mesmo número de *breaks* (i.e., séries de confrontos disputados sucessivamente em casa ou fora de casa). Outros objetivos incluem aumentar o número de confrontos transmitidos pela TV, minimizar o número de *breaks*, criar calendários em que as preferências específicas das equipes são levadas em consideração e minimizar a distância total viajada por todos as equipes durante o campeonato, entre vários outros.

O último objetivo citado é um problema de otimização conhecido como *Traveling Tournament Problem* (TTP) definido por Easton et al. [2001]. No TTP, os campeonatos são normalmente do tipo DRR e duas equipes não podem se enfrentar duas vezes consecutivas, ou seja, em rodadas consecutivas. O TTP é um problema de otimização $\mathcal{NP}$-Difícil no sentido forte, conforme demonstrado em Thielen e Westphal [2011].

Em campeonatos em que as equipes têm suas sedes uma longe das outras faz-se necessário a realização de várias viagens ao longo do campeonato. Isso gera custos com o transporte das delegações, além de impactar fisicamente os atletas, influenciando o rendimento dos mesmos durante as partidas. Assim, o TTP se mostra extremamente importante para a minimização da distância viajada pelas equipes durante os campeonatos e para a equilíbrio das condições de disputa.

Uma variação do TTP é o *Mirrored Traveling Tournament Problem* (mTTP) definido por Ribeiro e Urrutia [2007]. O mTTP é também um problema de otimização $\mathcal{NP}$-Difícil e é aplicado em campeonatos DRR espelhados (ou *Mirrored Double Round Robin*, MDRR) com n equipes, em que n é um número par. Cada equipe deve enfrentar duas vezes cada uma das outras equipes: uma vez no primeiro turno (ou primeiro SRR) e a segunda vez no segundo turno (ou segundo SRR). Duas equipes não devem se enfrentar mais do que uma vez em cada turno. A sequência de adversários de cada equipe deve ser a mesma tanto no primeiro turno quanto no segundo turno, porém, os HAPs (sede dos confrontos) devem ser invertidos no segundo turno.

Dadas n equipes, sendo n um número par, o total de rodadas de um campeonato MDRR é $r = 2(n-1)$, sendo $n-1$ rodadas no primeiro turno e outras $n-1$ rodadas no segundo turno. Em cada rodada devem existir $\frac{n}{2}$ confrontos, sendo que cada equipe deve possuir apenas um confronto por rodada. Confrontos entre duas equipes E_i e E_j são definidos por pares ordenados (i, j). As distâncias d_{ij} entre as cidades das equipes i e j são conhecidas e cada equipe começa o campeonato em sua respectiva cidade. Quando uma equipe possui uma sequência de confrontos fora de casa essa equipe vai de uma cidade de um adversário para outra cidade de outro adversário sem retornar para casa, i.e., a equipe sai em turnê. A tabela gerada pelo mTTP deve obedecer às seguintes restrições:

https://proceedings.science/p/85467?lang=pt-br

1. Nenhuma equipe pode ter mais do que três confrontos em sequência em casa ou três confrontos em sequência fora de casa;

2. Um confronto entre uma equipe E_i e uma equipe E_j que aconteceu na casa de E_i não pode ser seguido de outro confronto entre E_i e E_j na casa de E_j. Em outras palavras, não pode existir dois confrontos consecutivos entre duas equipes;

3. Os confrontos em uma rodada R são exatamente os mesmos confrontos em uma rodada $R + (n - 1)$ para $R = 1, 2, 3, \ldots, n - 1$, com o local dos confrontos invertidos.

A Tabela 1 apresenta a matriz de distâncias D para a instância NL4. Esta é uma instância derivada de dados reais da liga americana de beisebol profissional dos Estados Unidos e Canadá que contém quatro equipes: *Atlanta Braves* (ATL), *New York Mets* (NYM), *Philadelphia Phillies* (PHI) e *Montreal Expos* (MON).

Tabela 1: Exemplo de matriz de distâncias D da instância NL4.

Equipe	ATL	NYM	PHI	MON
ATL	0	745	665	929
NYM	745	0	80	337
PHI	665	80	0	380
MON	929	337	380	0

Um exemplo de HAPs para cada equipe da instância NL4 é mostrado na Tabela 2. Nos HAPs apresentados, C indica um confronto disputado em casa, e F indica um confronto disputado fora. No caso da equipe ATL, o HAP indica os três primeiros confrontos como mandante e os três últimos confrontos como visitante.

Tabela 2: Exemplo de HAPs para a instância NL4.

Equipe	Primeiro Turno			Segundo Turno		
ATL	C	C	C	F	F	F
NYM	C	F	F	F	C	C
PHI	F	C	C	C	F	F
MON	F	F	F	C	C	C

Uma solução para o mTTP consiste em uma tabela MDDR de tamanho $n \cdot 2(n - 1)$ que representa quais equipes se enfrentam e o local do enfrentamento em cada rodada. A Tabela 3 apresenta uma possível solução para a instância NL4, com base nos HAPs apresentados anteriormente. O sinal positivo indica que a equipe da coluna será a visitante e o sinal negativo indica que o enfrentamento é fora de casa.

Tabela 3: Exemplo tabela de confrontos para instância NL4.

	Primeiro Turno			Segundo Turno		
Equipe	Rodada 1	Rodada 2	Rodada 3	Rodada 4	Rodada 5	Rodada 6
ATL	PHI	NYM	MON	-PHI	-NYM	-MON
NYM	MON	-ATL	-PHI	-MON	ATL	PHI
PHI	-ATL	MON	NYM	ATL	-MON	-NYM
MON	-NYM	-PHI	-ATL	NYM	PHI	ATL

https://proceedings.science/p/85467?lang=pt-br

O custo produzido por cada equipe em uma solução é determinada pela distância percorrida por essa equipe durante o campeonato desde a primeira vez que essa equipe deixa sua cidade até a última vez que retorna. A partir da matriz de distâncias D apresentada na Tabela 1 e da tabela de confrontos apresentada na Tabela 3, pode-se calcular por exemplo, o custo total produzido pela equipe ATL, que é dado por:

$$d_{11} + d_{11} + d_{11} + d_{13} + d_{32} + d_{24} = 1082$$

Para o cálculo do custo total de todas as equipes na função objetivo uma variável binária x_{ijk} é utilizada:

$$x_{ijk} = \begin{cases} 1, & \text{se a equipe } E_i \text{ confronta a equipe } E_j \text{ na rodada } k \\ 0, & \text{caso contrário} \end{cases} \tag{1}$$

A função objetivo é dada pela Equação (2), que visa minimizar a total distância viajada por todas as equipes do campeonato:

$$\min\ Z_{mTTP} = \sum_{k=0}^{r} \sum_{j=0}^{n} \sum_{i=0}^{n} x_{ijk} d_{ij} \tag{2}$$

Este trabalho propõe a adaptação e aplicação da metaheurística Algoritmo Genético de Chaves Aleatórias Viciadas (*Biased Random-Key Genetic Algorithm*, BRKGA) em conjunto com o método do polígono à resolução do mTTP. São consideradas instâncias benchmark da literatura e os resultados obtidos são comparados com os resultados disponíveis.

As próximas seções deste trabalho são organizadas como segue. Na Seção 2 é apresentada a revisão da literatura sobre o mTTP. A Seção 3 descreve o funcionamento do BRKGA, do método do polígono e os detalhes da aplicação destes métodos ao mTTP. Os experimentos computacionais são apresentados e discutidos na Seção 4. Por fim, a Seção 5, apresenta as conclusões a respeito do trabalho produzido e aponta futuras direções de pesquisa.

2. Revisão da Literatura

Motivado pelo problema de se organizar o calendário da principal liga americana de beisebol (*Major League Baseball*), o *Traveling Tournament Problem* (TTP) foi concebido por Easton et al. [2001] com o objetivo de se minimizar a distância total viajada pelas equipes durante o torneio. O TTP tem se mostrado um grande desafio para a pesquisa operacional por se tratar de um problema de otimização $\mathcal{NP}$-Difícil no sentido forte, conforme demonstrado em Thielen e Westphal [2011].

Uma das primeiras metaheurísticas de bom desempenho para a solução do TTP foi apresentada por Anagnostopoulos et al. [2003], denominada *Traveling Tournament Simulated Annealing* (TTSA). O TTSA foi a primeira metaheurística a explorar vizinhanças cujos os movimentos incluíam a troca de equipes, troca de rodadas e a troca de confrontos através da tabela. O TTSA ajusta dinamicamente a função objetivo para balancear o tempo gasto em regiões viáveis e inviáveis, se assemelhando a uma estratégia de oscilação em busca tabu. O TTSA também utiliza um mecanismo de reaquecimento para escapar de mínimos locais em baixas temperaturas. Este mecanismo aumenta a temperatura novamente e divide a busca em várias fases.

Baseado no TTSA, Van Hentenryck e Vergados [2006] apresentaram extensões e melhorias para solucionar o caso em que o segundo turno possui a mesma sequência de confrontos porém, com a sede dos confrontos invertida. Foram considerados novamente os mesmos movimentos utilizados por Anagnostopoulos et al. [2003] e foram adicionadas pequenas modificações que levavam em consideração as restrições de espelhamento da tabela. Todos os melhores resultados até então foram igualados ou superados por este novo método, exceto por uma instância.

https://proceedings.science/p/85467?lang=pt-br

Outra aplicação do *simulated annealing* para solucionar o TTP foi desenvolvida em Lim et al. [2006]. O *simulated annealing* foi utilizado para gerar os HAPs e um algoritmo de *hill-climbing* foi utilizado para atribuir as equipes aos HAPs gerados. Primeiramente as soluções iniciais eram geradas pela metaheurística *beam search*. Com o intuito de se reduzir o tempo computacional e gerar soluções viáveis para um grande número de equipes, os padrões gerados nessa etapa inicial eram espelhados. Assim, o número de HAPs possíveis a serem gerados foi reduzido e deste modo a solução também era espelhada, embora esse não fosse o foco inicial.

Posteriormente, Ribeiro e Urrutia [2007] formalizaram a variante espelhada do TTP, denominada *Mirrored Traveling Tournament Problem*, embora esta já fosse conhecida da comunidade acadêmica. Considera-se este trabalho como o primeiro a se concentrar especificamente em instâncias espelhadas do TTP. Foi também apresentada uma heurística híbrida que combina os princípios do *Greedy Randomized Adaptive Search Procedures* (GRASP) e do *Iterated Local Search* (ILS), denominada GRILS-mTTP. Esta heurística obteve bons resultados tanto para instâncias espelhadas quanto para não espelhadas. Adicionalmente, esta metaheurística serviu como base para algumas das outras heurísticas futuras da literatura.

Di Gaspero e Schaerf [2007] utilizaram as vizinhanças propostas por Anagnostopoulos et al. [2003] e uma nova adicional para produzir diferentes combinações de vizinhanças. Nestas combinações, alguns movimentos que se sobrepunham, gerando redundâncias na busca por uma solução, foram descartados. Assim, o espaço de busca foi reduzido. As combinações de vizinhanças criadas foram analisadas e comparadas. Posteriormente, estas foram embutidas em uma busca tabu a fim de se solucionar o TTP.

Uma das vizinhanças estudadas pelos trabalhos mencionados é a *troca parcial de equipes*. A busca local por troca parcial de equipes é frequentemente utilizada para resolver problemas de *round robin sports scheduling* como o TTP e o mTTP. Foi realizada uma análise específica da conectividade desta vizinhança por Januario et al. [2016]. Concluiu-se que a conectividade é baixa, o que torna difícil se movimentar entre soluções vizinhas diferentes. Os métodos propostos, baseados nesta análise, incluem método do polígono e método *Faro Shuffle*. Adicionalmente, foram caracterizados os casos em que a busca local por troca parcial de equipes é particularmente útil.

Biajoli e Lorena [2006] utilizaram um Algoritmo Genético em conjunto com o método do polígono para solução do mTTP. O método do polígono foi utilizado para gerar a tabela a partir de um cromossomo codificado como uma permutação. Após a aplicação de operadores de as mutação e *crossover*, o *simulated annealing* foi utilizado para encontrar ótimos locais.

Outro método evolutivo foi utilizado por Gupta et al. [2013] em que o *Biogeography Based Optimization* (BBO) foi utilizado como processo evolutivo. Neste artigo, o método do polígono e o *simulated annealing* também são utilizados de maneira similar ao trabalho de Biajoli e Lorena [2006]. Ambos os trabalhos tiveram resultados parecidos e que não trouxeram grandes melhorias em relação aos melhores resultados disponíveis emde suas respectivas épocas.

Dois modelos de Programação Linear Inteira foram apresentados por de Carvalho e Lorena [2012], que utilizaram o conceito de grafos de conflitos. Cada nó no grafo representava um confronto e se esses nós fossem adjacentes, os respectivos confrontos não poderiam ocorrer na mesma rodada devido ao fato de serem conflitantes. Adicionalmente, foi introduzida uma nova versão do TPP, denominada MinMaxTTP cujo objetivo era a minimização da maior distância viajada pelas equipes.

3. Métodos Propostos

Este trabalho une o BRKGA ao método do polígono e utiliza diferentes métodos de busca local, inspirados pelas aplicações anteriores na literatura. O BRKGA é responsável por gerar soluções e permutações codificadas por meio de chaves aleatórias viciadas e o método do polígono é responsável pela decodificação dessas chaves transformando-as em soluções para o mTTP. Cada solução é submetida a cinco diferentes buscas locais, no intuito de acelerar o processo evolutivo.

https://proceedings.science/p/85467?lang=pt-br

3.1. Algoritmo Genético de Chaves Aleatórias Viciadas

Algoritmos Genéticos (AG) são uma classe de metaheurísticas populacionais evolutivas, comumente usadas para solucionar problemas de otimização combinatória. São métodos inspirados na teoria da evolução de *Darwin*, de forma que cada *individuo* ou *cromossomo* representa uma solução para um problema de otimização específico, normalmente dividido em porções de informação denominadas *genes*. Uma população evolui ao longo de gerações através de mecanismos de intensificação e diversificação das soluções, representados por operadores de reprodução (ou *crossover*) e mutação, respectivamente. Cada indivíduo é avaliado de acordo com sua aptidão (ou *fitness*), de maneira que os mais aptos têm maior chance de perpetuarem seu material genético ao longo das gerações. Ao final da execução de um AG, o indivíduo mais apto é a melhor solução encontrada em todas as gerações.

Existem as mais diversas variações dos AGs. Dentre elas, há os Algoritmos Genéticos de Chaves Aleatórias (*Random-Key Genetic Algorithm*, RKGA) introduzido por Bean [1994]. Neste método, cada individuo é codificado como um vetor de chaves aleatórias, em que cada chave é um numero real pertencente ao intervalo [0,1). Novamente, os indivíduos representam soluções para um determinado problema, entretanto, é necessário um decodificador para transformar o vetor de chaves aleatórias em uma solução de fato para o problema tratado, além de determinar sua aptidão. No RKGA, uma porcentagem dos indivíduos mais aptos de cada geração pertencem a um conjunto denominado *elite*. A cada geração no RKGA, uma nova população é criada, composta pelo conjunto elite da geração anterior, por indivíduos gerados pela reprodução de indíviduos da geração anterior e por indivíduos mutantes. Particularmente, o RKGA utiliza a reprodução uniforme parametrizada de Spears e De Jong [1995], na qual o i-ésimo gene do cromossomo *filho* gerado pode receber a i-ésima chave de um dos dois cromossomos *pai*, selecionados aleatoriamente, com a mesma probabilidade. O operador de mutação consiste em introduzir novos indivíduos, cujos genes são gerados de maneira aleatória.

Uma variação do RKGA foi recentemente introduzida por Gonçalves e Resende [2011], denominada de Algoritmo Genético de Chaves Aleatórias Viciadas (*Biased Random-Key Genetic Algorithm*, BRKGA). O BRKGA segue os mesmos princípios do RKGA, diferindo principalmente quanto ao operador de reprodução. No BRKGA, os dois indivíduos pai que serão utilizados para gerar um cromossomo filho são novamente selecionados de maneira aleatória, entretanto, um dos indivíduos deve pertencer ao conjunto elite e o outro deve pertencer ao restante da população. Adicionalmente, embora seja utilizado a reprodução uniforme parametrizada, os genes advindos do indivíduo pai do conjunto elite possuem maior probabilidade de serem atribuídos ao individuo filho, em detrimento do individuo pai não pertencente ao conjunto elite.

Para aplicação ao mTTP, o BRKGA gera vetores de chaves aleatórias de tamanho equivalente ao número *n* de equipes que participam do torneio. O intuito é gerar permutações de tamanho n, que posteriormente são transformadas em tabelas SRR pelo método o polígono. Para obter tais permutações, a decodificação de cada indivíduo consiste em ordenar o vetor de chaves aleatórias de cada indivíduo de forma não decrescente de acordo com o valor das chaves, mantendo junto com a chave o seu índice original. Assim, obtém-se permutações dos n índices de cada indivíduo.

3.2. Método do Polígono

O método do polígono [Dinitz et al., 1995] é comumente utilizado na literatura, sendo utilizado por exemplo por Biajoli e Lorena [2006] e por Gupta et al. [2013]. Com o método do polígono é possível, a partir de uma permutação base dos índices das equipes, representar uma tabela SRR com todos os confrontos e todas as rodadas. A permutação base, corresponde à primeira rodada da tabela do campeonato. Conforme mencionado, neste trabalho a permutação base é obtida por meio da decodificação dos indivíduos do BRKGA. A Tabela 4 apresenta um exemplo de permutação base para uma instância que contém seis equipes.

https://proceedings.science/p/85467?lang=pt-br

Tabela 4: Permutação base para seis equipes.

5	4	6	2	3	1

Considerando os dados da Tabela 4 é possível criar uma tabela SRR que corresponde às cinco rodadas de um torneio. Inicialmente, a permutação base é expandida, e posteriormente, os confrontos são definidos baseados nesta expansão.

A expansão consiste em gerar n-1 permutações. A primeira consiste da própria permutação base. Para geração das demais permutações, fixa-se o primeiro elemento da permutação base na primeira posição de todas as demais. O segundo elemento de cada permutação é o terceiro elemento da permutação anterior, o terceiro elemento é o quarto elemento da permutação anterior e assim consecutivamente até o último elemento, que é o segundo elemento da permutação anterior. Para o exemplo da permutação da Tabela 4 o resultado da expansão gerada pelo método do polígono é apresentado na Tabela 5.

Tabela 5: Expansão gerada pelo método do polígono.

5	4	6	2	3	1
5	6	2	3	1	4
5	2	3	1	4	6
5	3	1	4	6	2
5	1	4	6	2	3

A expansão gerada pelo método do polígono é utilizada como base para geração de uma tabela SRR. Considerando a expansão da Tabela 5, a tabela SRR apresentada na Tabela 6, na qual cada linha corresponde a uma equipe e cada coluna corresponde a uma rodada, é gerada como descrito a seguir. Para cada rodada $Ri = 1, ..., n-1$, a equipe da posição 1 da permutação i confronta a equipe na posição 2 da mesma permutação. De maneira análoga, as equipes das posições $i = 3, ..., (n/2) + 1$ confrontam as equipes das posições $n - i + 3$. Considerando a permutação base [5, 4, 6, 2, 3, 1], temos a definição da rodada 1 com os confrontos das equipes das posições 1 e 2 (5×4); 3 e 6 (6×1); e 4 e 2 (2×3), o que corresponde à primeira coluna da Tabela 6.

Tabela 6: Tabela SRR obtida relativa ao primeiro turno.

	R_1	R_2	R_3	R_4	R_5
E_1	6	3	4	2	5
E_2	3	4	5	1	6
E_3	2	1	6	5	4
E_4	5	2	1	6	3
E_5	4	6	2	3	1
E_6	1	5	3	4	2

Ainda é necessário associar os HAPs aos confrontos apresentados na Tabela 6. Para este fim, é utilizada a heurística gulosa proposta em Ribeiro e Urrutia [2007]. De acordo com este método, os HAPs da primeira rodada são gerados aleatoriamente. Para as rodadas seguintes, cada confronto é analisado e o mando de campo é determinado pela equipe envolvida com maior número de confrontos consecutivos anteriores em casa ou fora. Com base nesta informação, o padrão predominante é invertido, e.g., se uma equipe vem de confrontos consecutivos anteriores como mandante, o confronto atual será disputado como visitante. Caso as duas equipes tenham o mesmo número de confrontos consecutivos anteriores com mesmo padrão, tenta-se inverter o padrão da mesma maneira, ou, caso seja impossível, o mando de campo é gerado aleatoriamente.

https://proceedings.science/p/85467?lang=pt-br

Determinados os HAPs, o segundo turno é criado espelhando-se o primeiro. O resultado final é uma tabela MDRR, como a apresentada na Tabela 7.

Tabela 7: Tabela MDRR contendo os confrontos dos dois turnos.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
E_1	-6	3	-4	2	-5	6	-3	4	-2	5
E_2	-3	4	-5	-1	6	3	-4	5	1	-6
E_3	2	-1	6	5	-4	-2	1	-6	-5	4
E_4	5	-2	1	-6	3	-5	2	-1	6	-3
E_5	-4	6	2	-3	1	4	-6	-2	3	-1
E_6	1	-5	-3	4	-2	-1	5	3	-4	2

3.3. Buscas Locais

Após a fase de decodificação, as tabelas SRR são submetidas a cinco buscas locais diferentes, propostas por Anagnostopoulos et al. [2003]. Cada tabela gerada é submetida a cada busca local com probabilidade $\rho \in [0, 1)$. Todas as buscas locais são aplicadas com política de primeira melhoria, de maneira que movimentos de piora não são aceitos. A seguir, cada busca local é descrita brevemente.

3.3.1. *Home-Away Swap* - HAS

Nesta busca local um movimento consiste em, dado um confronto entre duas equipes E_i e E_j, inverter o mando de campo nos confrontos entre essas duas equipes. Considerando o exemplo da Tabela 7, um movimento consiste em multiplicar o valor de qualquer entrada na tabela por -1. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local.

3.3.2. *Round Swap* - RS

Dadas duas rodadas R_i e R_j ($i \neq j$), esta busca local consiste em realizar a troca dos confrontos entre as duas rodadas na tabela. Considerando o exemplo da Tabela 7, um possível movimento consiste em trocar as colunas R_1 com R_2 e consequentemente R_6 com R_7, com isso, altera-se a distância viajada pelas equipes. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local.

3.3.3. *Team Swap* - TS

Dadas duas equipes E_i e E_j, um movimento desta busca local consiste em trocar todos os confrontos da equipe E_i com os confrontos da equipe E_j, exceto o confronto entre as duas. O mando de campo no confronto entre E_i e E_j é mantido o mesmo. Considerando o exemplo da Tabela 7, um possível movimento consiste em trocar as linhas E_1 e E_6, exceto pelas colunas R_1 e R_6, correspondentes aos confrontos entre os times correspondentes. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local.

3.3.4. *Partial Round Swap* - PRS

Dadas duas rodadas R_i e R_j ($i \neq j$) e uma equipe E_k. Um movimento desta busca local consiste em trocar a equipe que E_k enfrenta na rodada R_i com a equipe que E_k enfrenta na rodada R_j. Este movimento pode tornar uma solução inviável, de maneira que é necessário criar uma cadeia de ejeção incluindo outros confrontos para manter a viabilidade da solução. Considerando o exemplo da Tabela 7, um possível movimento consiste em trocar os elementos (E_1, R_1), (E_1, R_2). Por se tratar da versão espelhada do problema, os elementos (E_1, R_6), (E_1, R_7) também devem ser trocados. Entretanto, isso implicaria na violação da restrição de apenas um jogo por rodada pela equipe 3 nas rodadas R_1 e R_6 e pela equipe 6 nas rodadas R_2 e R_7, e modificações adicionais seriam necessárias para manter a viabilidade da solução. Existe um total de $O(n^3)$ movimentos possíveis nesta busca local.

https://proceedings.science/p/85467?lang=pt-br

3.3.5. *Partial Team Swap* - PTS

Dadas duas equipes E_i e E_j ($i \neq j$) e uma rodada R_k, um movimento desta busca local consiste em trocar a equipe que E_i enfrenta na rodada R_k com a equipe que E_j enfrenta na rodada R_k. Assim como na busca local PRS, são necessárias outras modificações na tabela para que a solução se mantenha viável. Considerando o exemplo da Tabela 7, um possível movimento consiste em trocar os elementos (E_1, R_5), (E_6, R_5). Por se tratar da versão espelhada do problema, os elementos (E_1, R_{10}), (E_6, R_{10}) também devem ser trocados. Entretanto, isso implicaria na violação da restrição de apenas um jogo por rodada novamente, exigindo modificações adicionais para manter a viabilidade da solução. Existe um total de $O(n^3)$ movimentos possíveis nesta busca local.

3.4. Correção dos Cromossomos

Cada busca local altera de alguma forma a tabela gerada pelo método do polígono a partir da decodificação do cromossomo. Após a aplicação das buscas locais, torna-se necessário atualizar o cromossomo para que a representação da solução seja fidedigna.

Entretanto, nem todas as alterações realizadas podem ser codificadas como uma única permutação. Por este motivo, optou-se por uma representação indireta parcial, de maneira que a permutação que mais se aproxima da tabela gerada pelas buscas locais é utilizada para atualizar as chaves do cromossomo após a aplicação das buscas locais. As informações eventualmente perdidas neste processo podem ser alcançadas novamente por meio de uma nova aplicação das buscas locais.

3.5. Visão Geral

O esquema geral do BRKGA é mostrado no fluxograma apresentado na Figura 1. Nela é possível verificar o fluxo de execução e o relacionamento entre os diferentes componentes utilizados.

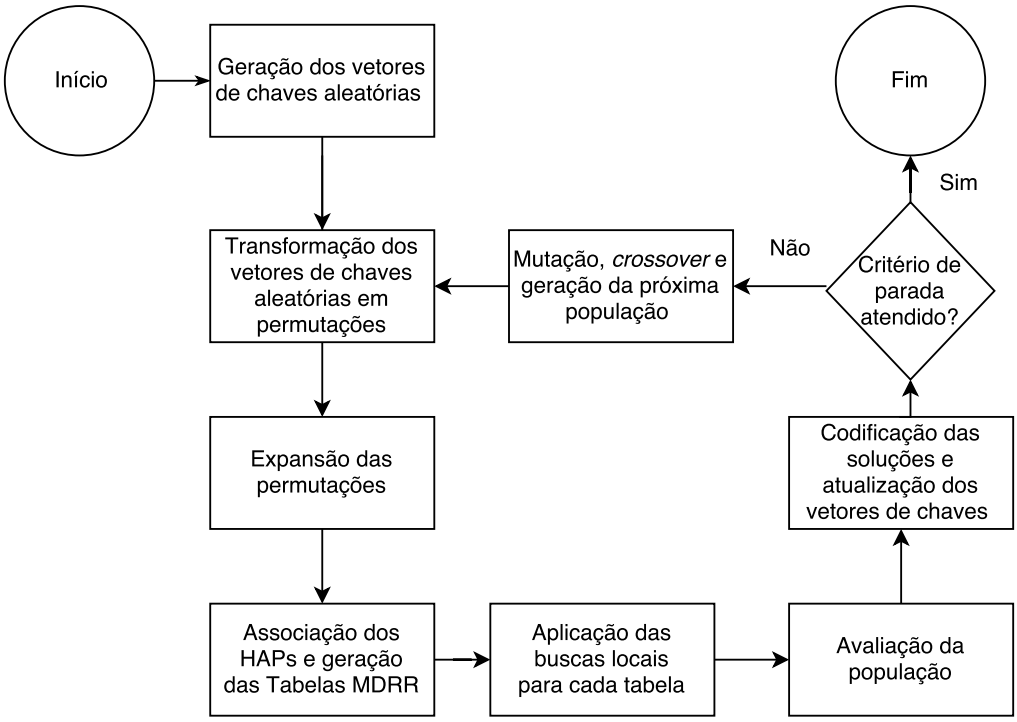


Figura 1: Fluxograma da abordagem proposta.

4. Experimentos Computacionais

Os experimentos computacionais foram realizados em um computador Intel Core i5 Quad Core de 3.0 GHz com 8 GB de RAM sob o sistema operacional Ubuntu 15.10. Foi utilizada a *brkgaAPI*, desenvolvida em C++ por Toso e Resende [2015], contendo as implementações de todas as fases do BRKGA. O código original foi alterado para adaptação ao mTTP, compilado com g++ 5.2.1 e opção de otimização -O3 foi utilizada.

https://proceedings.science/p/85467?lang=pt-br

4.1. Ajuste de Parâmetros

Para o ajuste da melhor configuração dos parâmetros, foi utilizado o pacote *irace* proposto por López-Ibáñez et al. [2016]. O *irace* é uma ferramenta que possui procedimentos de configuração automática utilizando métodos estatísticos e de corrida iterada. Além dos parâmetros do BRKGA também foi ajustada a ordem em que as buscas locais são aplicadas. O número máximo de gerações, tamanho da população e o ρ utilizado nas buscas locais foram ajustados manualmente verificando o tempo de execução médio para cada instância. A melhor configuração das buscas locais encontrada pelo *irace* foi a sequência RS, TS, PRS, PTS e por último, HAS. Os demais parâmetros são apresentados na Tabela 8.

Tabela 8: Parâmetros definidos para o BRKGA.

Parâmetro	Valor Definido
Tamanho da população	$3\times n$
Tamanho do conjunto elite	25%
Tamanho do conjunto mutante	20%
Probabilidade de herdar gene elite na reprodução	85%
Número de gerações	800
ρ	80%

4.2. Comparação com Resultados da Literatura

As instâncias utilizadas e as referências dos valores das melhores soluções estão disponíveis na seção de instâncias espelhadas no *site* oficial do pesquisador Michael Trick `http://mat.gsia.cmu.edu/TOURN/`. Para esse trabalho foram consideradas as instâncias da *National League* (NL) e as instâncias circulares (*circ*). Devido ao uso de componentes de aleatoriedade, o método proposto foi executado independentemente 10 vezes para cada instância.

As Tabelas 9 e 10 apresentam os resultados para cada instância considerada. São apresentados o valor da melhor solução conhecida na literatura (B^*), o melhor valor da solução obtida pelo BRKGA (S^*), o valor médio das soluções (S), o tempo de execução médio em segundos (T), a geração média de convergência das soluções ($conv$), a distância percentual entre os resultados (gap) obtidos e os melhores valores das soluções para cada instância e o desvio padrão (σ) em relação ao valor médio das soluções. Os melhores resultados da literatura são devidos a diferentes métodos, não havendo um método específico que reúna todos os melhores resultados. A distância percentual é calculada como $gap = \frac{S^*-B^*}{B^*} \times 100$.

Tabela 9: Resultados para instâncias NL.

Instância	B^*	S^*	S	T (s)	$conv$	gap (%)	σ
nl4	8276	8276	8276,00	0,55	5,4	0,00	0,00
nl6	26588	26588	26588,00	3,30	28,3	0,00	0,00
nl8	41928	43620	43743,10	12,04	447,7	4,04	139,43
nl10	63832	70386	71101,20	35,27	399,5	10,23	447,05
nl12	119608	130013	131540,70	86,84	363,6	8,70	962,41
nl14	199363	236461	239885,10	188,62	500,8	18,61	2212,51
nl16	278305	313555	317536,70	375,91	442,8	12,67	2776,91

De acordo com os dados reportados para as instâncias NL, o método proposto demonstra ser competitivo ao gerar soluções próximas àquelas geradas pelos melhores métodos presentes na literatura em baixo tempo computacional. Adicionalmente, o método demonstra consistência ao apresentar desvio padrão baixo, menor que 1% em execuções independentes. Por fim, o método

https://proceedings.science/p/85467?lang=pt-br

também demonstra boa convergência, atingindo as melhores soluções próximo à metade do número máximo de iterações.

Tabela 10: Resultados para instâncias circulares.

Instância	B^*	S^*	S	T (s)	$conv$	gap (%)	σ
circ4	20	20	20,00	0,55	0,00	0,00	0,00
circ6	72	72	72,00	3,21	0,56	0,00	0,00
circ8	140	150	151,80	11,86	168,4	7,14	0,60
circ10	272	300	302,80	34,62	370,10	10,29	2,71
circ12	432	496	506,60	85,52	387,10	14,81	4,10
circ14	672	756	784,20	187,87	377,00	12,50	11,81
circ16	968	1102	1107,60	374,26	299,2	13,84	4,63
circ18	1306	1570	1582,89	846,08	373,89	20,21	9,10
circ20	1852	2134	2163,33	1461,20	305,77	15,23	18,26

Os resultados gerados para as instâncias circulares possuem qualidade similar aos obtidos para as instâncias NL. Como este conjunto possui maiores instâncias é possível verificar um aumento no tempo de execução para as instâncias com 18 e 20 equipes. A convergência e o desvio padrão são menores para este conjunto de instâncias, ao passo que o *gap* médio apresentou leve aumento.

5. Conclusões e Trabalhos Futuros

O *Mirrored Traveling Tournament Problem* (mTTP) é um problema $\mathcal{NP}$-Difícil que tem por objetivo otimizar a geração de tabelas de confrontos de torneios esportivos, minimizando a total distância viajada por todas as equipes durante o campeonato. Há ampla aplicação prática para métodos de solução deste problema, nos mais diferentes esportes praticados. Tabelas com distâncias viajadas reduzidas são imprescindíveis para minimizar os custos com transportes das equipes e aumentar o bem estar dos atletas.

Neste trabalho foi apresentada a abordagem ao mTTP aplicando-se a recente metaheurística Algoritmo Genético de Chaves Aleatórias Viciadas em conjunto com o método do polígono e cinco diferentes buscas locais. Os experimentos computacionais consideraram instâncias *benchmark* da literatura baseados em casos reais e compararam o método proposto às melhores soluções já obtidas por diferentes métodos da literatura. Os resultados obtidos demonstram que o método é competitivo, obtendo baixa divergência em relação às melhores soluções disponíveis para a maioria das instâncias em baixo tempo computacional. Adicionalmente, o método proposto apresenta rápida convergência e variação da qualidade das soluções abaixo de 1% em diferentes execuções independentes.

Os trabalhos futuros incluem testar diferentes esquemas de codificação do BRKGA e diferentes operadores de reprodução e mutação. Adicionalmente, serão consideradas modificações como o uso de operadores adaptativos e controle de diversidade do conjunto elite.

6. Agradecimentos

Esta pesquisa foi apoiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico, pela Fundação de Apoio à Pesquisa do Estado de Minas Gerais e pela Universidade Federal de Ouro Preto.

Referências

Anagnostopoulos, A., Michel, L., Van Hentenryck, P., e Vergados, Y. (2003). A simulated annealing approach to the traveling tournament problem. In *Journal of Scheduling*. Citeseer.

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160.

https://proceedings.science/p/85467?lang=pt-br

Biajoli, F. e Lorena, L. (2006). Mirrored traveling tournament problem: An evolutionary approach. *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, p. 208–217.

de Carvalho, M. A. M. e Lorena, L. A. N. (2012). New models for the mirrored traveling tournament problem. *Computers & Industrial Engineering*, 63(4):1089–1095.

Di Gaspero, L. e Schaerf, A. (2007). A composite-neighborhood tabu search approach to the traveling tournament problem. *Journal of Heuristics*, 13(2):189–207.

Dinitz, J., Lamken, E., e Wallis, W. (1995). Scheduling a tournament. *Handbook of Combinatorial Designs*, p. 578–584.

Easton, K., Nemhauser, G., e Trick, M. (2001). The traveling tournament problem description and benchmarks. In *International Conference on Principles and Practice of Constraint Programming*, p. 580–584. Springer.

Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Gupta, D., Goel, L., e Aggarwal, V. (2013). A hybrid biogeography based heuristic for the mirrored traveling tournament problem. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, p. 325–330. IEEE.

Januario, T., Urrutia, S., e de Werra, D. (2016). Sports scheduling search space connectivity: A riffle shuffle driven approach. *Discrete Applied Mathematics*, 211:113–120.

Lim, A., Rodrigues, B., e Zhang, X. (2006). A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3):1459–1478.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Ribeiro, C. C. e Urrutia, S. (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3):775–787.

Spears, W. M. e De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. Technical report, Naval Research Lab, Washington D.C.

Thielen, C. e Westphal, S. (2011). Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4-5):345–351.

Toso, R. F. e Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93.

Van Hentenryck, P. e Vergados, Y. (2006). Traveling tournament scheduling: a systematic evaluation of simulated annealling. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, p. 228–243. Springer.

https://proceedings.science/p/85467?lang=pt-br