Um método baseado em Estratégia Evolutiva para a resolução do Problema de Agendamento em Competições Esportivas

Alexandre J. T. Maciel Filho[1], Fernando Bernardes de Oliveira[2], Rafael F. Alexandre[3]
Departamento de Computação e Sistemas
Universidade Federal de Ouro Preto (UFOP)
CEP: 35931-008 – João Monlevade – MG – Brasil
alexandrej.maciel@gmail.com[1], fernando@decea.ufop.br[2],
rfalexandre@decea.ufop.br[3]

RESUMO

O Problema de Agendamento em Competições Esportivas envolve muitos fatores que fazem a competição acontecer, como interesses econômicos, competitividade das equipes, além dos ganhos de patrocinadores. O objetivo deste trabalho é propor uma meta-heurística baseada em Estratégia Evolutiva para este tipo de problema, tendo como objetivo minimizar o tempo médio percorrido pelas equipes durante a competição. O Campeonato Brasileiro de Futebol foi adotado como estudo de caso, em virtude das características do problema, além do interesse nacional para o assunto. Foram definidos alguns operadores para a manipulação de soluções com o propósito de atender as restrições impostas pela Confederação Brasileira de Futebol. Considerando o contexto experimental, os resultados sugerem um desempenho satisfatório do algoritmo. Quando comparado com as tabelas oficiais do Campeonato Brasileiro dos anos de 2014, 2015 e 2016, o algoritmo proposto encontrou menor tempo total de viagem para os times.

PALAVRAS CHAVE. Competições Esportivas, *Round Robin*, Meta-heurística, Estratégias Evolutivas.

Tópicos: OC – Otimização Combinatória; MH – Meta-heurísticas.

ABSTRACT

There are several factors related to Sports Competitions Problems, as economic interesting, competitiveness of teams, besides the sponsor gains. This work proposes a metaheuristic based on Evolution Strategy to deal with this class of problem. The objective is to minimize the average time spent for teams during competition. The Brazilian Football Championship was studied because of its properties and addition to the national interest to that issue. Some operators were defined to respect constraints imposed by the Brazilian Football Confederation. For the experimental context, the results suggest the proposed algorithm has a good performance. When it was compared with official tables from 2014, 2015 and 2016, the method found a smaller total travel time for teams.

KEYWORDS. Sports Competitions. Round Robin. Metaheuristic. Evolution Strategy.

Paper topics: Combinatorial optimization; Metaheuristic.

1. Introdução

As ligas desportivas profissionais existem em todo o mundo e, em muitos casos, elas são populares e de enorme importância econômica. O volume monetário envolvido com a venda de bilhetes e direitos de transmissão para os jogos é muito alto e, assim, o planejamento dessas ligas é de grande importância. Um aspecto importante é a geração de um calendário para os torneios, a qual determina a ordem e o local em que as equipes deverão jogar entre si durante a temporada. Esse tipo de problema é formalmente definido por Easton et al. [2001] e denominado *Traveling Tournament Problem* (TTP). Algumas variantes do modelo de torneio são conhecidas como *round robin* simples, no qual todas as equipes jogam uma contra as outras apenas uma única vez, e *round robin* duplo, quando existem dois jogos entre cada uma das equipes do torneio. Sendo conhecidas as equipes participantes do campeonato e a distância entre as cidades de cada uma delas, o objetivo do TTP é minimizar a distância percorrida pelas equipes durante a temporada. Como as distâncias deslocadas pelas equipes para a realização dos jogos podem ser grandes, ocasionando um grande desgaste dos atletas com as viagens, a resolução desse tipo de problema torna-se importante.

O futebol no Brasil é um dos esportes mais importantes, com enorme impacto social e econômico, além de envolver milhões de pessoas. Para isso, os campeonatos precisam ser competitivos para que continue atraindo fãs de todo o Brasil. O Campeonato Brasileiro de Futebol é um evento criado e administrado pela Confederação Brasileira de Futebol (CBF) e, além disso, atrai diversos fãs que sempre acompanham seus times durante todo andamento. É um campeonato extenso, geralmente iniciado no mês de maio e finalizado em dezembro. Além disso, equipes de várias regiões do Brasil percorrem grande parte do país para realizar os jogos. Com intuito de que o campeonato tenha um cronograma competitivo e mantenha os fãs dos times com expectativa a cada rodada, deve-se criar o calendário seguindo restrições determinadas para que nenhum time tenha vantagem sobre outros. O Campeonato Brasileiro utiliza o modelo *round robin* duplo, e tem como critério de classificação os pontos corridos. O time que conseguir maior pontuação consagra-se campeão. No Campeonato Brasileiro de Futebol da Série A participam 20 times que jogam entre si em turno e returno, sendo 19 jogos no primeiro turno e 19 jogos no segundo turno. Cada time pode jogar uma única vez na rodada, e quando os times se enfrentam no primeiro turno, eles enfrentam novamente na mesma rodada do segundo turno com mando de campo invertido.

A proposta deste trabalho é definir uma meta-heurística, baseada em técnicas de computação evolucionária, para o problema de geração da tabela do Campeonato Brasileiro de Futebol, em especial para a Série A. O objetivo é definir os jogos de maneira que o tempo médio percorrido pelas equipes seja minimizado. O problema consiste em alocar os jogos, agrupados em turno e returno, para as equipes da série em questão. Cabe ressaltar que o algoritmo proposto pode ser aplicado a qualquer campeonato que possui o formato *round robin*, e que o Campeonato Brasileiro de Futebol foi escolhido como estudo de caso devido às características do problema, além do interesse nacional para o assunto.

O restante deste trabalho está organizado como segue. A Seção 2 descreve os aspectos mais relevantes do Campeonato Brasileiro de Futebol, bem como as restrições impostas pela CBF. A Seção 3 define as características do algoritmo proposto neste trabalho e, também, os operadores especialmente desenvolvidos para manipular as soluções candidatas para o problema. O contexto experimental e os resultados obtidos com o algoritmo proposto são discutidos na Seção 4. As conclusões e as propostas de trabalhos futuros são apresentadas na Seção 5.

2. Formulação do problema

Esta seção apresenta uma breve revisão da literatura discutindo os principais trabalhos correlatos e suas características. A seguir, questões acerca dos critérios identificados para a formulação do problema serão discutidas. Após isso, a função objetivo definida para o problema será apresentada. Cabe ressaltar que as principais características envolvidas na definição do problema abordam questões que podem influenciar diretamente no desgaste das equipes como, por exemplo, o tempo de deslocamento entre as partidas.

2.1. Trabalhos correlatos

Dentre as abordagens definidas para o problema, Biajoli [2003] utilizou meta-heurísticas por meio do algoritmo *Simulated Annealing* e Busca Tabu. O objetivo do autor é a minimização dos gastos financeiros com o deslocamento das equipes durante o campeonato. Resultados relevantes foram apresentados reduzindo a distância média percorrida entre os times quando comparada com a tabela oficial.

Bartsch et al. [2006] utilizaram três heurísticas para criar um calendário que cumpria os requisitos estabelecidos para as ligas Alemã e Australiana. O resultado do trabalho gerou um calendário utilizável para as ligas. As heurísticas desenvolvidas foram baseadas em *Semi-greedy*, *Truncated branch-and-bound* e *Exact branch-and-bound* para criar um calendário para liga alemã e, para a liga australiana, utilizou a heurística de *Semi-greedy*. Os resultados sugerem que foram criados alguns calendários com baixo tempo computacional.

Rasmussen e Trick [2006] utilizaram Programação Inteira (PI) para solução de um modelo restrito, além de um algoritmo *branch and price* para resolver o problema de minimização de distância dos jogos de torneios do modelo *round robin*. A utilização do algoritmo hibrido mostrou bons resultados, e foi definido no contexto como melhor método utilizado para o problema. Já o algoritmo de *Branch and Price* não obteve resultados computacionais tão bons quanto o outro.

Ribeiro e Urrutia [2012] abordaram o problema por meio de Programação Linear Inteira. Foram determinados requisitos pela CBF e pela emissora que detém os direitos de transmissão dos jogos. As propostas eram maximizar o número de públicos nos estádios, ter uma maior audiência para TV e criar um torneio mais equilibrado. De acordo com Ribeiro [2012], essa solução foi utilizada no Campeonato Brasileiro de 2009. O campeonato naquele ano foi considerado o mais atraente, pois o título foi disputado até a última rodada por quatro times.

A partir de métodos de Programação Inteira, Alarcón et al. [2017] elaboraram um calendário para a Associação Chilena de Futebol Profissional. O objetivo era permitir um campeonato competitivo, sendo atraente para os fãs mantendo os custos. Os autores conseguiram aumentar o interesse pelo campeonato, além de aumentar a receita e diminuir os custos de viagem dos times.

Carlsson et al. [2017] propuseram um calendário para principal liga Sueca de Handball utilizando *Constraint Programming*. Além das diversas restrições que eram impostas pela liga, foi considerado um modelo genérico para o campeonato definido como Duplo *Round Robin* com divisões de jogos. Segundo aqueles autores, os resultados apresentam melhorias sobre as abordagens que utilizam Programação Inteira e decomposição, além de reduzir o tempo de execução.

2.2. Critérios para a construção da tabela

A definição da tabela de um campeonato de maneira adequada, atendendo as restrições e não privilegiando nenhuma equipe, é uma tarefa muito difícil. Quando um time percorre uma distância maior que os outros times, sugere-se que ele tenha um desgaste maior e, além disso, maior custo com despesas de viagem. Entende-se que isso pode ocasionar num rendimento menor na competição quando comparado aos times que percorreram menores distâncias. Segundo Agência Futebol Interior [2015], para minimizar esse desgaste seria adequado que a diferença das distâncias percorridas entre os times fosse a menor possível.

O critério a ser considerado neste trabalho como objetivo a ser otimizado é o tempo de viagem das equipes. Observou-se que o contexto mais adequado é a minimização do tempo médio de deslocamento que, por sua vez, está diretamente associado a distância percorrida pelos times. De modo geral, quanto maior a distância percorrida pelo time, maior é o tempo de viagem.

Outro fator que foi levado em consideração é existem times que participam do campeonato que não possuem aeroportos comerciais em sua cidade. Sendo assim, as equipes viajam de avião até uma cidade próxima e terminam a viagem de ônibus. Em alguns casos, os times podem ainda fazer toda a viagem de ônibus, o que pode não ser viável pela longa distância entre as cidades. Essas viagens de ônibus costumam ser mais demoradas, aumentando o desgaste dos atletas que passariam mais tempo na estrada. Por exemplo, isso pode ser observado na Tabela 1, em que a distância

Origem	Destino	Distancia em Linha reta	Tempo
Belo Horizonte	São Paulo	491,40 km	1h15min
Belo Horizonte	Santos	514,26 km	2h45min

Tabela 1: Distância e tempo aproximado de Belo Horizonte a Santos e São Paulo.

de Belo Horizonte para Santos e de Belo Horizonte para São Paulo são parecidas. Entretanto, existe uma diferença de tempo considerável entre essas cidades. Como Santos não possui aeroporto comercial, uma opção para chegar até lá partindo de Belo Horizonte seria ir de avião para São Paulo e, logo após, prosseguir de ônibus até Santos.

Considerando as questões acerca do deslocamento, a função objetivo desse trabalho foi definida a partir do somatório do tempo necessário para as equipes deslocarem dos locais onde se encontram até o local da partida. Com a redução do tempo total deslocado espera-se um menor desgaste das equipes. As considerações sobre essa função, bem como as restrições utilizadas são discutidas a seguir.

2.3. Função objetivo e restrições

A função objetivo deste trabalho, a qual deve ser minimizada, é representada pela seguinte equação:

$$Min \sum_{i,j,k} T_{ij} L_{ijk} \tag{1}$$

Na Equação 1, i e j representam as equipes e k representa a rodadas. As funções T e L representam o tempo de deslocamento entre as equipes i e j, considerando o local de cada equipe em relação à rodada anterior. Isso define se a equipe no jogo anterior estava em sua cidade sede ou em outra cidade.

Tratando especificamente do Campeonato Brasileiro de Futebol, a CBF define um conjunto de regras específicas que devem ser atendidas para que a tabela do campeonato seja viável. Segundo aquela entidade, essas restrições têm como objetivo deixar o campeonato mais competitivo.

As restrições consideradas neste trabalho foram baseadas no trabalho de Ribeiro e Urrutia [2012], e são apresentadas a seguir. Para facilitar o entendimento, as restrições serão categorizadas em quatro grupos: A) Restrições Gerais; B) Padrão Casa *versus* Fora; C) Clássicos ou Jogos Regionais; e D) Restrições Geográficas.

A.1: Cada equipe deve jogar contra as outras equipes por duas vezes, sendo uma vez em casa e outra fora.

A.2: Cada equipe deve jogar apenas uma vez em cada rodada, seja em casa ou fora (cronograma compacto).

A.3: Cada equipe deve jogar contra outra equipe exatamente uma vez em cada turno, ao longo do $n-1$ rodadas. Os jogos do segundo turno são reproduzidas exatamente na mesma ordem do primeiro turno, entretanto, com mando de campo invertido.

B.1: As equipes alternam o mando de campo (*casa* ou *fora*) em cada rodada durante as primeiras quatro rodadas do turno e as duas últimas. Em outras palavras, não há jogos seguidos nas quatro primeiras ou nas duas últimas rodadas do campeonato.

B.2: Equipes que jogam em casa na primeira rodada necessariamente jogam fora na segunda, e vice-versa.

B.3: Cada equipe joga 19 jogos por turno, sendo que jogará em cada turno um jogo a mais em casa ou fora.

B.4: Cada equipe só pode jogar duas vezes em casa e/ou fora em rodadas consecutivas.

C.1: Nenhum jogo considerado como "clássico" poderá ocorrer nas três primeiras rodadas e, também, jogos regionais/clássicos nas últimas quatro rodadas do campeonato.

C.2: Não pode haver mais que um jogo "clássico" ocorrendo na mesma cidade em qualquer rodada.

C.3: Nenhuma equipe pode jogar dois jogos clássicos em rodadas consecutivas.

D.1: Cada equipe deve jogar um jogo fora do estado em que a sua cidade de origem está localizado em qualquer uma das duas primeiras rodadas.

As restrições estabelecidas pela empresa detentora dos direitos de imagem dos jogos, as quais descrevem questões acerca do horário e dia que os jogos devem acontecer, não foram contempladas neste trabalho. Restrições Geográficas que limitam a quantidade de jogos seguidos nos estados não foram consideradas neste trabalho também, pois não foi observado como vantagem jogar mais de 5 jogos no mesmo estado.

3. Algoritmo Proposto

O algoritmo proposto nesse trabalho é baseado na meta-heurística Estratégia Evolutiva (μ + λ) (EE, do inglês *Evolution Strategy*), a qual é uma classe de algoritmos evolutivos. Os passos principais para esse método podem ser definidos como: [Engelbrecht, 2007; Luke, 2013]

1. Criação da população inicial: um número μ de indivíduos gerados aleatoriamente.
2. Reprodução: cada indivíduo irá gerar λ/μ descendentes por meio de mutação.
3. Competição: os descendentes irão competir com os pais por meio de seus *fitness*.
4. Seleção: para a próxima geração serão selecionados os μ indivíduos mais aptos do conjunto formado por (μ + λ) indivíduos.

Algoritmo 1: Algoritmo ES($\mu + \lambda$)

Entrada: μ, λ, $maxGen$
Saída : $melhorInd$

```
1  gen ← 0;
2  P ← geraPopInicial ();
3  melhorInd ← atualizaMelhor (P);
4  enquanto gen < maxGen faça
5      D ← {} ; // Descendentes
6      para cada indivíduo i ∈ P faça
7          para λ/μ faça
8              j ← criaDescendente (i, p_m);
9              avaliarSolucao (j);
10             inserirSolucao (D, j);
11         fim
12     fim
13     definirSobreviventes (P, D, μ, λ) ; // ES (μ + λ)
14     melhorInd ← atualizaMelhor (P);
15     gen ← gen + 1;
16 fim
17 retorne melhorInd
```

O algoritmo proposto possui operadores que realizam a reparação de algumas restrições violadas durante a execução da mutação. O objetivo é minimizar o tempo de viajem dos times do Campeonato Brasileiro, conforme definido pela Equação 1. O EE foi a estratégia escolhida por utilizar como operador principal a mutação. O processo seria mais complexo se fosse utilizado algum método que emprega o *crossover*, pois sua aplicação poderia implicar na violação de diversas restrições. A representação de uma solução é apresentada na Seção 3.1. Os operadores desenvolvidos serão descritos nas Seção 3.3. Os passos de execução são apresentados no Algoritmo 1.

A algoritmo recebe o tamanho da população μ, o número de descentes gerados λ para cada indivíduo e o máximo de gerações $maxGen$. A população inicial é gerada (linha 2) conforme os critérios estabelecidos na Seção 3.2. Na linha 3, o melhor indivíduo da população é recuperado. Enquanto o critério de parada não for atingido, considerando neste caso o número máximo de gerações, os seguintes passos são executados (linhas 4–16). Uma lista vazia de descendentes é criada (linha 5). Para cada indivíduo i da população P (linha 6), λ/μ descendentes são criados (linha 7) observando as seguintes ações. O descendente j é criado (linha 8) a partir do indivíduo i por meio dos operadores de mutação (Seção 3.3). Na linha 9, o descendente gerado é avaliado. Após isso, a solução é inserida na lista de descendentes (linha 10). Após todos os descendentes serem criados, os sobreviventes são definidos (linha 13). As populações P e D são unidas e classificadas conforme o *fitness*, e apenas μ indivíduos são selecionados. Os demais são descartados. O melhor indivíduo corrente é atualizado (linha 14), e o número de gerações é incrementado (linha 15). Ao final do processo (linha 17), o melhor indivíduo é retornado.

3.1. Representação da solução

Cada solução da população é definida por uma matriz M que representa a tabela do campeonato, e um vetor V contendo N posições, no qual são armazenadas as informações dos times. Assim, a representação completa de cada solução pode ser expressa por $S = \{M, V\}$, sendo que M é uma matriz com $2N - 2$ linhas (rodadas) e $N/2$ colunas (jogos). O segundo turno do campeonato consiste no primeiro turno (matriz M), com os mandos de campo invertidos.

O Campeonato Brasileiro atual é composto por 38 rodadas, divididas em dois turnos de 19 rodadas. Cada uma dessas rodadas possui 10 jogos (20 times), e cada célula da matriz $M_{i,j}$ representa um jogo entre os times. Por exemplo, no formato do campeonato atual com 20 times, um fragmento de uma tabela resultado é ilustrado pela Figura 1.

Vetor de times (V)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T

Representação da tabela de jogos (matriz M)

AxB	CxD	ExF	GxH	IxJ	KxL	MxN	OxP	QxR	SxT
NxC	RxI	HxM	DxE	FxG	TxQ	JxO	PxA	LxS	BxK
KxD	MxF	QxJ	AxT	CxR	IxH	GxL	OxN	SxB	ExP
⋮									
BxD	MxJ	ExI	LxT	RxN	GxA	FxH	SxO	KxC	QXP
JxG	BxQ	FxA	DxO	NxK	PxI	LxE	HxS	TxC	RxM
AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP

Mapeamento dos times do vetor V

A -> Flamengo	K -> Botafogo
B -> Sport	L -> São Paulo
C -> Plameiras	M -> Figueirense
D -> Atlético-PR	N -> Ponte Preta
E -> Atlético-MG	O -> Corinthians
F -> Santos	P -> Grêmio
G -> Coritiba	Q -> América-MG
H -> Cruzeiro	R -> Fluminense
I -> Santa Cruz	S -> Internacional
J -> Vitória	T -> Chapecoense

Figura 1: Representação de uma tabela em que 20 times do Campeonato Brasileiro são mapeados para os identificadores A a T. A cada identificador é atribuída a uma posição do vetor V. A matriz M ilustra possíveis jogos para o primeiro turno do campeonato.

O vetor de 20 posições contém todas informações dos times participantes do campeonato brasileiro atual. Para efeito de ilustração, os confrontos são separados por um ”X”. O time que estiver ao lado esquerdo será o time que jogará em seu estádio, ou seja, o mandante do jogo (casa). O time que estiver ao lado direito será o visitante (fora). Sendo assim, a Figura 1 ilustra o jogo Flamengo contra Sport (jogo 1 da rodada 1 – AxB), Palmeiras contra Atlético Paranaense (jogo 2 da rodada 1 – CxD), e assim por diante. Cada time armazenado no vetor contém um conjunto de informações sendo elas: i) identificação; (ii) Nome; iii) Cidade; iv) Estado; v) Localização atual; vi) Tempo de viagem gasto e; vii) Lista com a identificação de Clássicos.

3.2. Geração da população inicial

Uma tabela do Campeonato Brasileiro, como o exemplo apresentado na Figura 1, será utilizada como modelo para construção da população inicial com μ indivíduos. Cada solução é construída a partir dos times embaralhados e, assim, tendo seus jogos alterados aleatoriamente. Logo,

para cada solução gerada, os times podem estar associados a diferentes letras do vetor V. Devido ao embaralhamento dos times, existe a possibilidade de haver soluções infactíveis na população inicial construída.

Em um primeiro momento, a estrutura da matriz M não é alterada. A operação para construir diferentes estruturas de matrizes é discutida na Seção 3.3.

3.3. Operadores utilizados

O algoritmo utiliza três tipos de operadores de mutação, sendo possível selecionar as combinações entre eles. Os operadores são chamados de: i) Troca de times com reparo de clássicos nas três primeiras rodadas utilizando times de estados diferentes (Seção 3.3.1); ii) Troca de times com reparo de clássicos nas três primeiras rodadas mudando times entre os clássicos (Seção 3.3.2); e iii) Troca de rodadas com reparo de jogos seguidos mudando mando de campo (Seção 3.3.3). Estes operadores são aplicados no primeiro turno e espelhado para o segundo turno. A seguir, cada um dos operadores é descrito.

3.3.1. Troca de times com reparo de clássicos nas três primeiras rodadas utilizando times de estados diferentes

Este operador é executado em duas etapas. Na primeira, os novos indivíduos são criados a partir da troca de times. Isso é feito selecionando dois times aleatórios e trocando suas partidas umas pelas outras. Por exemplo, as partidas do time A são trocadas pelas partidas do time B, sendo que um time assume as partidas do outro.

Após essa operação, a solução é avaliada para verificar se existem clássicos nas três primeiras rodadas. Por exemplo, se houver um jogo entre Cruzeiro e Atlético em umas das três primeiras rodadas, inicia-se o processo de reparo. Um dos dois times que disputa o clássico será selecionado aleatoriamente, e seus jogos serão trocados por um time de um estado diferente também selecionado aleatoriamente. Essa etapa é realizada até que não existam mais clássicos nas três primeiras rodadas. Este procedimento também não altera a estrutura da matriz M.

3.3.2. Troca de times com reparo de clássicos nas três primeiras rodadas mudando times entre os clássicos

Este operador é executado em duas etapas. A primeira etapa é realizada da mesma maneira que o procedimento anterior, e segue os mesmos critérios estabelecidos na Seção 3.3.1. Por exemplo, os times A e B foram selecionados aleatoriamente, e seus jogos são trocados. Esse processo é ilustrado na Figura 2.

Figura 2: Troca de times aleatórios. Existe a troca entre o time A e B, representados por Sport e Flamengo, respectivamente.

Após essa operação, a solução é avaliada para verificar se existem clássicos nas três primeiras rodadas. Este procedimento fará a troca entre os jogos resultantes utilizando os times que não formam clássicos entre si. Na Figura 2, após a troca de A e B, ocorrem os clássicos entre B e K. Foi identificado também um clássico entre G e L na terceira rodada. O procedimento

consiste em selecionar um dos times que disputa o primeiro clássico e outro time que disputa o segundo clássico, desde que também não formem clássicos, e seus jogos serão trocados. Caso o procedimento seja executado por 5 vezes sem conseguir resolver o problema dos clássicos, um time de cada clássico é selecionado aleatoriamente, e trocado por um time de um estado diferente. A operação é realizada até que não existam mais clássicos nas três primeiras rodadas.

3.3.3. Troca de rodadas com reparo de jogos seguidos mudando mando de campo

Este operador é executado em duas etapas. Na primeira etapa, os novos indivíduos são criados a partir da troca de rodadas. Duas rodadas são escolhidas aleatoriamente entre as rodadas 5 e 17, e seus jogos são trocados. Por exemplo, se forem selecionadas as rodadas 7 e 13, todos os jogos da rodada 7 irão acontecer na rodada 13 e os da rodada 13 irão acontecer na rodada 7.

Após essa troca de rodadas, é verificado se algum time joga mais de dois jogos seguidos em casa ou fora de casa. Caso essa regra seja violada, avalia-se o erro a partir da rodada 4. Seleciona-se aleatoriamente a rodada 5 ou 6, e inverte-se o mando de campo do jogo do time que infringiu a regra. A rodada 4 não pode ser alterada pelo fato de que nas quatro primeiras rodadas os times devem jogar com o mando de campo invertidos. Se o erro não aconteceu com a rodada 4, o mando de campo do jogo do time que infringiu a regra é invertido. Após a troca de mando de campo, verifica-se se existem mais times com jogos seguidos. A operação é realizada até que não existam mais jogos seguidos, conforme as restrições.

4. Experimentos computacionais

O algoritmo foi desenvolvido na linguagem de programação *Java* utilizando o *framework jMetal* (*Metaheuristic Algorithms in Java*) [Durillo e Nebro, 2011]. Este *framework* incorpora diversas meta-heurísticas utilizadas em problemas de otimização, tanto no contexto mono-objetivo quanto multiobjetivo. A partir da reutilização do código, permite que diversos métodos sejam implementados, pois vários algoritmos utilizam os mesmos componentes básicos, como os operadores, busca local, cálculo da *fitness*, entre outros. Para a implementação foi utilizada a plataforma *NetBeans* IDE 8.2.

4.1. Ambiente computacional e parametrização

Todas as execuções do algoritmo foram realizadas em um notebook com as seguintes especificações: Sistema Operacional *Windows* 7, processador Intel(R) *Core*(TM) i7-2630QM 2.00 GHz e 8 GB de memória RAM. Durante o processamento, apenas o ambiente de desenvolvimento era utilizado, e as demais aplicações estavam fechadas. A ordem de execução dos testes foi aleatorizada.

Testes preliminares foram realizados para a definição dos parâmetros a serem utilizados no experimento. A partir da variação número de gerações e dos parâmetros μ e λ, foi possível realizar a definição de valores que permitem uma convergência satisfatória do algoritmo. Os parâmetros adotados para execução do algoritmo podem ser observados na Tabela 2. O número máximo de gerações foi utilizado como critério de parada para o algoritmo.

Tabela 2: Tabela de parâmetros

Parâmetro	**Valor**
Gerações	20.000
μ	40
λ	$3\mu = 120$

4.2. Resultados e análises

Para avaliar a performance do algoritmo proposto, foi realizado um experimento considerando dois operadores de mutação. Estes operadores foram definidos a partir de testes preliminares, observando que os demais geravam inconsistências na tabela por violar algumas restrições.

Como instâncias de teste foram utilizadas as seguintes tabelas: Campeonato Brasileiro 2016 (CB2016), Campeonato Brasileiro 2015 (CB2015) e Campeonato Brasileiro 2014 (CB2014), sendo todas da Série A. As instâncias foram geradas a partir das informações obtidas das Tabelas Oficiais dos campeonatos de cada ano divulgadas pela CBF. O tempo estimado para as viagens aéreas foram calculados com base na distância entre os aeroportos das cidades das equipes que se enfrentarão. Para as equipes que não possuem aeroportos em suas cidades-sedes, considerou-se o tempo estimado para a viagem aérea acrescido da distância rodoviária até o aeroporto mais próximo da cidade-sede. Por exemplo, o deslocamento da equipe do Santos até o aeroporto em São Paulo. Os tempos estimados de viagem para o deslocamento aéreo foram obtidos por meio do *site*[1]. Já para o deslocamento rodoviário, os tempos estimados foram obtidos a partir do *site*[2].

As combinações dos operadores de mutação que foram empregados são as seguintes:

CO.1 Operador de troca de rodadas com reparo de jogos seguidos mudando mando de campo (Seção 3.3.3) juntamente com operador de troca de times com reparo de clássicos nas três primeiras rodadas utilizando times de estados diferentes (Seção 3.3.1);

CO.2 Operador de troca de rodadas com reparo de jogos seguidos mudando mando de campo (Seção 3.3.3) juntamente com operador de troca de times com reparo de clássico nas três primeiras rodadas mudando times entre os clássicos (Seção 3.3.2).

Foram realizadas 33 execuções para cada combinação de operadores. O tempo médio de execução para cada uma das execuções foi de 4 minutos. Na Tabela 3 é apresentado o resultado das execuções para cada uma das combinações utilizadas. Na tabela foram inseridos o menor resultado (*Menor*), o qual corresponde ao melhor valor encontrado pelo algoritmo; o maior resultado (*Maior*), o qual representa ao indivíduo factível com maior valor encontrado pelo algoritmo; e o valor médio (*Média*) dos indivíduos em cada um dos contextos. Para representar a dispersão dos dados e avaliar a variabilidade dos resultados, foi calculado também o desvio padrão (*Desvio*).

Tabela 3: Resultado da combinação dos operadores

Operadores	**CO1**				**CO2**			
Instância	Menor	Maior	Média	Desvio	Menor	Maior	Média	Desvio
CB2016	**73.717**	75.957	74.860,90	556,77	74.053	75.600	74.941,61	351,59
CB2015	**77.912**	80.184	78.978,12	354,00	78.262	80.164	79.249,71	268,36
CB2014	86.106	87.121	86.674,81	268,19	**85.812**	87.128	85.587,55	321,97

Para avaliar a diferença entre os resultados foi utilizado o cálculo do GAP, definido pela Equação 2. Esse cálculo representa a relação, em percentual, do desempenho entre os operadores.

$$GAP = \frac{CO1 - CO2}{CO1} * 100 \quad (2)$$

Os resultados obtidos pelas execuções sugerem que, para a instância CB2016, o operador CO.1 foi 0,45% melhor que a CO.2. Para a instância CB2015, o operador CO.1 foi 0,44% melhor e na instância CB2014, o operador CO.2 foi 0,34% melhor. Mesmo os resultados sendo aparentemente próximos, a diferença de tempo entre eles pode ser significativa, considerando que isso pode influenciar no desgaste das equipes. A diferença entre os resultados também pode ser observada pelos *boxplots* apresentados na Figura 3. É possível observar que o valor máximo está mais afastado dos valores centrais, e que utilizando o CO.2 a dispersão dos dados é menor nas tabelas de 2015 e 2016. Entretanto, para este operador em 2016, o desempenho médio é maior em relação ao CO.1.

[1] `http://decolar.com/`
[2] `http://www.distanciascidades.com/`

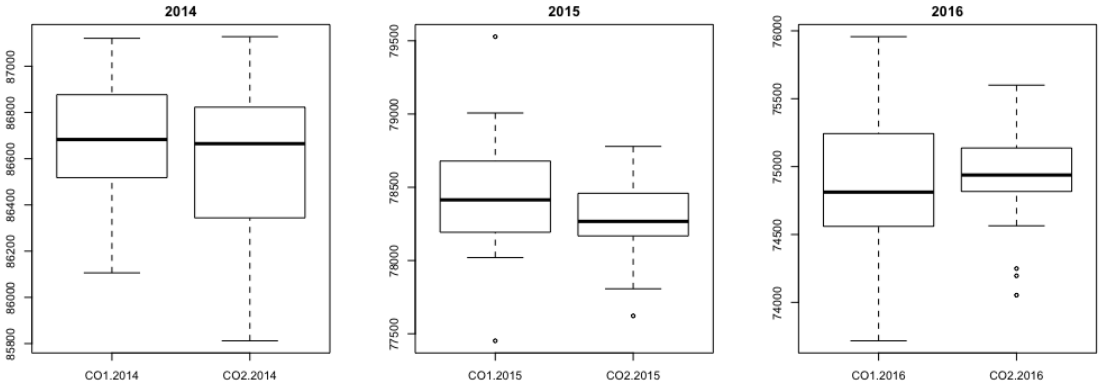

Figura 3: *Boxplots* dos resultados.

Com os resultados apresentados pelo algoritmo na Tabela 4, é possível também realizar a comparação do melhor resultado apresentado com os valores obtidos a partir Tabela Oficial do campeonato nos respectivos anos. Foram consideradas as restrições observadas neste trabalho e tendo como critério o tempo total de viagens dos times. Essa comparação é apresentada na Tabela 4, na qual é mostrado o menor resultado (*Menor*), a diferença do tempo gasto do time (*DTGT*) que mais passou tempo viajando com o time que viajou menos. Para a Tabela Oficial divulgada pela CBF é apresentado o *fitness*, conforme o cálculo realizado pela Equação 1, e a diferença DTGT.

Os valores destacados em negrito representam os melhores valores encontrados em cada uma das instâncias. Os resultados sugerem que o algoritmo conseguiu encontrar tanto um tempo total menor quanto uma diferença entre os times que têm deslocamento menor e maior. Isso pode sugerir que o desgaste das equipes pode ser menor, além de tornar a diferença de deslocamento mais justa e o campeonato mais competitivo.

Tabela 4: Resultado da combinação dos operadores

Operador	**CO1**		**CO2**		**Tabela Oficial**	
Instância	Menor	DTGT	Menor	DTGT	Fitness	DTGT
CB2016	**73.717**	**2.836**	74.053	2.877	80.991	3.631
CB2015	**77.452**	3.891	77.622	**3.613**	81.096	3.652
CB2014	86.106	3.998	**85.812**	**3.742**	89.532	4.219

Outro ponto a ser observado, conforme o ambiente experimental definido e as restrições impostas, que o algoritmo proposto foi capaz de gerar soluções viáveis e sem violar nenhuma restrição. Durante os cálculos para Tabela Oficial foi observado que pelo menos uma das restrições foi violada, de acordo com o contexto estabelecido por este trabalho na Seção 2.3. Isso pode ter acontecido em virtude de modificações realizadas para atender condições especiais, como possíveis questões impostas pela empresa detentora dos direitos de imagem dos jogos. Entretanto, tais situações estão fora do escopo deste trabalho, e as violações foram consideradas apenas como ilustração para o problema.

Na Tabela de 2016, a diferença do total de tempo do melhor resultado do conjunto de operadores CO1 para a Tabela Oficial é de 7.274 minutos. Conforme o cálculo do GAP, isso corresponde à uma diferença de 9%, aproximadamente. Foi observado que essa tabela viola a restrição D1, na qual é definido que cada equipe deve jogar fora de seu estado em uma das duas primeiras rodadas.

Para a Tabela de 2015, a diferença do total de tempo do melhor resultado do operador CO1 para a Tabela Oficial é de 3.644 minutos. Conforme o cálculo do GAP, isso corresponde à

uma diferença de 4,5%, aproximadamente. Foi observado que essa tabela também viola a restrição D1.

Já na Tabela de 2014, o melhor resultado foi encontrado com o conjunto de operadores CO2. A diferença do total de tempo do melhor resultado desse conjunto para a Tabela Oficial é de 3.720 minutos. Conforme o cálculo do GAP, isso corresponde à uma diferença de 4,2%, aproximadamente. Foi observado que essa tabela não atendeu duas restrições estabelecidas. A primeira delas, $B1$, se referem que nas primeiras quatro rodadas os jogos devem ser alternados (casa e fora). A outra, $C1$, define que não podem ocorrer clássicos nas três primeiras rodadas e nem regionais ou clássicos nas quatro ultimas.

De acordo com as condições experimentais e o contexto deste trabalho, na maioria dos testes realizados o algoritmo conseguiu gerar resultados com tabelas viáveis. As restrições violadas foram apresentar para indicar que o algoritmo conseguiu atender ao escopo estabelecido.

É importante observar que o algoritmo foi capaz de obter melhora no tempo de viagem total e na diferença do time que mais viaja para o time que menos viaja, conforme o ambiente experimental estabelecido. As diferenças conforme o GAP em relação à Tabela Oficial e o conjunto de operadores que encontrou o melhor resultado são aproximadamente: 2016 (CO1): 22%; 2015 (CO2): 1% e 2014 (CO2): 11%. Os resultados sugerem uma diferença significativa para os anos de 2016 e 2014, e um valor aproximado para 2015. Além disso, os resultados podem sugerir que a menor diferença pode resultar num campeonato mais competitivo.

De modo geral, os resultados sugerem que o algoritmo, mesmo partindo de uma solução inicial embaralhada, consegue obter uma solução factível diferente da Tabela Oficial utilizada pela CBF, por meio das combinações de operadores. Além disso, que o tempo pode ser melhor que a Tabela Oficial, considerando o contexto experimental determinado. Cabe ressaltar que o método proposto utiliza como critério de comparação as Tabelas Oficiais dos campeonatos. Sendo assim, a metodologia de validação do algoritmo proposto realiza uma comparação indireta com os diferentes métodos utilizados em cada ano. A CBF, entretanto, não divulgou as estratégias empregadas. Dessa maneira, considerando o contexto experimental apresentado, o método proposto conseguiu obter resultados satisfatórios e promissores.

5. Considerações Finais

Este trabalho teve como propósito apresentar um algoritmo baseado na meta-heurística Estratégia Evolutiva para construir tabelas de campeonatos que seguem o modelo *round robin* duplo. O objetivo estabelecido foi o de minimizar o tempo de viagem dos times que participam do campeonato. Como estudo de caso foi utilizado o Campeonato Brasileiro de Futebol, considerando o número de restrições envolvidas e por ser um campeonato longo. A minimização do tempo de viagem dos times é importante, pois pode ajudar a reduzir o desgaste das equipes.

O algoritmo proposto utiliza operadores especialmente desenvolvidos para trabalhar com a estrutura de soluções definidas nesse trabalho. Os operadores procuram modificar a solução corrente e reduzir o tempo de viagem das equipes. Considerando o contexto experimental, os resultados sugerem que o algoritmo alcançou seus objetivos utilizando as combinações de operadores estabelecidas. A redução no tempo de viagem pode ser observada das Tabelas do Campeonato Brasileiro dos anos de 2014, 2015 e 2016. Além disso, foi apresentada uma redução no tempo gasto do time que mais viajou para o time que menos viajou. As soluções geradas são viáveis porque não violaram nenhuma das restrições definidas neste trabalho.

Como propostas de continuidade e trabalhos futuros, sugere-se a implementação de uma versão multiobjetivo para o problema. Esse modelo pode atender, por exemplo, tanto as restrições dos horários da detentora de direitos de imagens dos jogos, bem como outras as restrições que não foram observadas nesse trabalho. No contexto multiobjetivo podem ser consideradas também a minimização da quantidade de jogos seguidos para cada time. Outra possibilidade é o estudo e a inclusão de métodos de busca local eficientes para melhorar os resultados obtidos pelo algoritmo

proposto e, além disso, a implementação de diferentes estratégias propostas na literatura para o problema do Campeonato Brasileiro, tornando possível a comparação com o método proposto. Além do contexto do Campeonato Brasileiro de Futebol, o algoritmo proposto pode ser adaptado para campeonatos que utilizem o mesmo conceito do campeonato e com regras próximas. A proposta é que as competições seja mais justo e dentro de todas as restrições que são importantes para um calendário competitivo.

Agradecimentos

Esta pesquisa foi apoiada pela Universidade Federal de Ouro Preto a partir do Auxílio Pesquisador/Pró-Reitoria de Pesquisa e Pós-Graduação – PROPP, e pela Fundação de Apoio à Pesquisa do Estado de Minas Gerais – FAPEMIG.

Referências

Agência Futebol Interior (2015). Sport viaja 3,6 vezes mais que trio de ferro e dá quase duas voltas ao mundo no brasileirão. http://m.futebolinterior.com.br/futebol/Brasileiro/Serie-A/2015/noticias/2015-07/Sport-viaja-3-6-vezes-mais-que-Trio-de-Ferro-no-Brasileirao. Acessado em: 20 de Abril 2016.

Alarcón, F., Durán, G., Guajardo, M., Miranda, J., Muñoz, H., Ramírez, L., Ramírez, M., Sauré, D., Siebert, M., Souyris, S., Weintraub, A., Wolf-Yadlin, R., e Zamorano, G. (2017). Operations research transforms the scheduling of chilean soccer leagues and south american world cup qualifiers. *Interfaces*, 47(1):52–69.

Bartsch, T., Drexl, A., e Kroger, S. (2006). Scheduling the professional soccer leagues of austria and germany. *Computers & Operations Research*, 33(7):1907 – 1937. ISSN 0305-0548. Special Issue: Operations Research in SportSpecial Issue: Operations Research in Sport.

Biajoli, F. L. (2003). Resolução do problema de jogos do campeonato brasileiro de futebol. Master's thesis, Departamento de Computação, Universidade Federal de Ouro Preto - UFOP, Ouro Preto, Brazil.

Carlsson, M., Johansson, M., e Larson, J. (2017). Scheduling double round-robin tournaments with divisional play using constraint programming. *European Journal of Operational Research*, 259 (3):1180 – 1190. ISSN 0377-2217.

Durillo, J. J. e Nebro, A. J. (2011). jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771. ISSN 0965-9978.

Easton, K., Nemhauser, G., e Trick, M. (2001). *The Traveling Tournament Problem Description and Benchmarks*, p. 580–584. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-45578-3.

Engelbrecht, A. (2007). *Computational Intelligence: An Introduction*. Wiley. ISBN 9780470512500.

Luke, S. (2013). *Essentials of Metaheuristics*. Lulu, 2 edition. Disponível em `https://cs.gmu.edu/~sean/book/metaheuristics/`.

Rasmussen, R. V. e Trick, M. A. (2006). *The Timetable Constrained Distance Minimization Problem*, p. 167–181. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-34307-3.

Ribeiro, C. C. (2012). Sports scheduling: Problems and applications. *International Transactions in Operational Research*, 19(1-2):201–226. ISSN 1475-3995. URL `http://dx.doi.org/10.1111/j.1475-3995.2011.00819.x`.

Ribeiro, C. C. e Urrutia, S. (2012). Scheduling the brazilian soccer tournament: Solution approach and practice. *Interfaces*, 42:260–272.