

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIENCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

ALEXANDRE JOSÉ TEIXEIRA MACIEL FILHO

**APLICAÇÃO DE UMA META-HEURÍSTICA PARA O PROBLEMA DE  
ALOCAÇÃO DE JOGOS DO CAMPEONATO BRASILEIRO DE FUTEBOL**

João Monlevade

2017

ALEXANDRE JOSÉ TEIXEIRA MACIEL FILHO

**Aplicação de uma meta-heurística para o problema de alocação de jogos do  
Campeonato Brasileiro de Futebol**

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Fernando Bernardes de Oliveira

Coorientador: Rafael Frederico Alexandre.

João Monlevade

2017



### ATA DE DEFESA

No quarto dia do mês de abril de 2017, às 16 horas e 30 minutos, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo aluno **Alexandre José Teixeira Maciel Filho**, sendo a Comissão Examinadora constituída pelos professores: Prof. Dr. Fernando Bernardes de Oliveira, Prof. Dr. Rafael Frederico Alexandre, Prof. Msc. Samuel Souza Brito, Prof. Msc. George Henrique Godim da Fonseca e Prof. Msc. Paganini Barcellos de Oliveira.

O candidato apresentou a monografia intitulada: "*Aplicação de uma meta-heurística para o problema de alocação de jogos do Campeonato Brasileiro de Futebol*". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,5 (Doze e Meio — 9 —), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final.

Na forma regulamentar foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo graduando.

João Monlevade, 04 de abril de 2017.

Prof. Dr. Fernando Bernardes de Oliveira  
Professor Orientador/Presidente

Prof. Dr. Rafael Frederico Alexandre  
Professor Coorientador

Prof. Msc. Samuel Souza Brito  
Professor Convidado

Prof. Msc. George Henrique Godim da Fonseca  
Professor Convidado

Prof. Msc. Paganini Barcellos de Oliveira  
Professor Convidado

Alexandre José Teixeira Maciel Filho  
Graduando



Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Colegiado do Curso de Sistemas de Informação  
Campus João Monlevade

**Folha de Aprovação**  
**Curso de Sistemas de Informação**

**FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA**

**APLICAÇÃO DE UMA META-HEURÍSTICA PARA O PROBLEMA DE ALOCAÇÃO DE  
JOGOS DO CAMPEONATO BRASILEIRO DE FUTEBOL**

**Alexandre José Teixeira Maciel Filho**

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:

Prof. Dr. Fernando Bernardes de Oliveira  
DECSI – UFOP

Prof. Dr. Rafael Frederico Alexandre  
DECSI – UFOP

Prof. Msc. Samuel Souza Brito  
DECSI – UFOP

Prof. Msc. George Henrique Godim da Fonseca  
DECSI – UFOP

Prof. Msc. Paganini Barcellos de Oliveira  
DEENP – UFOP

João Monlevade, 04 de abril de 2017.



UFOP

Universidade Federal  
de Ouro Preto

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

**ANEXO III – Termo de Responsabilidade**

**TERMO DE RESPONSABILIDADE**

Eu, Alexandre José Teixeira Maciel Filho,  
declaro que o texto do trabalho de conclusão de curso intitulado  
“ Aplicação de uma meta-Heurística para o problema de  
alocação de jogos do Campeonato Brasileiro de Futebol ” é de  
minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código  
fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas  
referências ou consentimento dos respectivos autores.

João Monlevade, 09 de Abril de 2012

Alexandre José Teixeira Maciel Filho  
Assinatura do aluno

## RESUMO

O problema de geração de tabelas para campeonatos envolve muitos fatores que fazem a competição acontecer, como interesses econômicos, competitividade das equipes além dos ganhos de patrocinadores. A geração de uma tabela imparcial é importante pois cria-se um campeonato disputado durante toda sua execução e como resultado disso atrai os fãs do esporte fazendo-os assistirem os jogos e, assim, gerando rendimento aos times. O objetivo desse trabalho é definir uma meta-heurística baseada em técnicas de computação evolucionária para o problema de geração da tabela do Campeonato Brasileiro de Futebol com intuito de minimizar o tempo médio percorrido pelas equipes. A meta-heurística utiliza o modelo de Estratégia Evolutiva. Foram definidos 5 operadores de mutação para criar tabelas que não infringissem as restrições impostas pela Confederação Brasileira de Futebol (CBF). Os resultados sugerem um desempenho satisfatório do algoritmo. Considerando o ambiente experimental, o algoritmo obteve um tempo total de viagens dos times menor do que os tempos contidos nas tabelas do Campeonato Brasileiro de 2014, 2015 e 2016, respeitando as restrições observadas, não violando nenhuma delas.

**Palavras-chave:** Meta-heurística; Geração de tabelas; Campeonato; Futebol; Estratégia Evolutiva

## **ABSTRACT**

The schedule generation problems for championships involve many factors that make a competition happen, as economic interest, the team's competitiveness, and the sponsors gains. The impartial schedule generation is important because it creates a competitive championship during all your execution, and as a result, it brings fans to the sport making them watch the games thus generating better performance of the teams. The goal of this job is to define a metaheuristic based on evolutionary computer techniques to the problem of schedule generation of the Brazilian Soccer Tournament in order to minimize the average time spent by teams. The metaheuristic uses the Evolutionary Strategy model. Five mutation operators were defined to create schedules that did not violate the restrictions imposed by the Brazilian Soccer Confederation (CBF). The results suggest a satisfactory performance of the algorithm. Considering the experimental environment, the algorithm surpassed the charts of the Brazilian Tournament of 2014, 2015 and 2016 in the observed restrictions since it did not violate any of them, and in the total time of trips of the teams being smaller than the official schedule.

**Keywords:** Metaheuristic; Schedule Generation; Tournament; Soccer; Evolutionary Strategy

## LISTA DE FIGURAS

Figura 1. Representação de uma tabela-----	23
Figura 2. Exemplo da estrutura do vetor de times antes do início dos jogos -----	24
Figura 3. Exemplo da estrutura do vetor de times na rodada 2-----	25
Figura 4. Tabela com times misturados aleatoriamente -----	25
Figura 5. Troca de times regionais-----	26
Figura 6. Troca de times aleatórios-----	27
Figura 7. Verifica se existe clássico -----	27
Figura 8. Troca um time dos clássicos por um time aleatório de outro estado-----	28
Figura 9. Troca de times aleatórios -----	29
Figura 10. Verifica se há clássicos nas 3 primeiras rodadas -----	29
Figura 11. Troca times entre clássicos da primeira rodada -----	30
Figura 12. Seleciona duas rodadas aleatórias e troca seus jogos. No exemplo foi selecionado para trocar as rodadas 7 e 13-----	31
Figura 13. Verifica se há algum time com mais de dois jogos seguidos em casa ou fora de casa. Nesse caso o time K tem 3 jogos seguidos em casa-----	31
Figura 14. Seleciona aleatoriamente um dos jogos. No caso foi selecionado o jogo KxT e invertido seus mandos de campo -----	32
Figura 15. Gráfico de convergência do algoritmo a partir de um valor factível -----	35
Figura 16. Boxplots dos resultados -----	36

## LISTA DE TABELAS

Tabela 1. Distância e tempo de Belo Horizonte até Santos e São Paulo-----	17
Tabela 2. Parâmetros -----	33
Tabela 3. Resultado da combinação dos operadores -----	35
Tabela 4. Comparação dos resultados obtidos com a tabela oficial -----	37

## LISTA DE ABREVIATURAS

CBF - Confederação Brasileira de Futebol-----	06
CMR - Criação de um Modelo de Restrições-----	20
DTGT - Diferença do Tempo Gasto do Time -----	36
EE - Estratégia Evolutiva-----	21
GNU - <i>Lesser General Public License</i> -----	33
PI - Programação Inteira -----	20

## LISTA DE ALGORITMOS

Algoritmo 1.1 - Procedimento Estratégia Evolutiva -----22

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 OBJETIVOS .....	15
1.2 ESTRUTURA DO TRABALHO .....	15
<b>2 FORMULAÇÃO DO PROBLEMA.....</b>	<b>16</b>
2.1 CRITÉRIOS PARA A CONSTRUÇÃO DA TABELA.....	16
2.2 FUNÇÃO OBJETIVO.....	17
2.2.1 RESTRIÇÕES DO TORNEIO .....	18
2.2.2 PADRÃO CASA-FORA .....	18
2.2.3 CLÁSSICOS OU JOGOS REGIONAIS .....	19
2.2.4 RESTRIÇÕES GEOGRÁFICAS.....	19
2.3 TRABALHOS CORRELATOS .....	19
2.4 CONSIDERAÇÕES FINAIS .....	20
<b>3 ALGORITMO PROPOSTO .....</b>	<b>21</b>
3.1 PSEUDOCÓDIGO DO ALGORITMO PROPOSTO.....	21
3.2 REPRESENTAÇÃO DAS SOLUÇÕES .....	23
3.3 GERAÇÃO DA POPULAÇÃO INICIAL.....	25
3.4 OPERADORES UTILIZADOS .....	26
3.4.1 TROCA DE TIMES REGIONAIS .....	26
3.4.2 TROCA DE TIMES COM REPARO DE CLÁSSICOS NAS 3 PRIMEIRAS RODADAS UTILIZANDO TIMES DE ESTADOS DIFERENTES .....	27
3.4.3 TROCA DE TIMES COM REPARO DE CLÁSSICOS NAS 3 PRIMEIRAS RODADAS MUDANDO TIMES ENTRE OS CLÁSSICOS.....	28
3.4.4 TROCA DE RODADAS COM REPARO DE JOGOS SEGUIDOS MUDANDO MANDO DE CAMPO .....	30
3.5 CONSIDERAÇÕES FINAIS .....	32

<b>4 RESULTADOS.....</b>	<b>33</b>
4.1 IMPLEMENTAÇÃO DO ALGORITMO.....	33
4.2 AMBIENTE COMPUTACIONAL E PARAMETRIZAÇÃO .....	33
4.3 RESULTADOS E ANÁLISES .....	34
4.1 CONSIDERAÇÕES FINAIS .....	38
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>39</b>
<b>6 REFERÊNCIAS.....</b>	<b>40</b>

## 1 Introdução

Campeonatos *round robin* utilizam um modo simples para criar seus torneios, separando seus jogos em pares e fazendo com que todos os times joguem entre si em um número de vezes em uma quantidade de rodadas. No *round robin* simples cada par de times se enfrenta apenas uma vez e no *round robin* duplo os times se enfrentam duas vezes. Quando o número de rodadas é mínimo e cada time joga apenas um jogo por rodada é chamado de *round robin* compacto. Também é considerado que cada equipe possua seu próprio estádio, assim, quando jogando em seu estádio essa equipe é denominada mandante e quando jogando fora de seu estádio ela é denominada visitante. A organização desses jogos deve ser feita para indicar em qual rodada o par de times deve jogar, definindo quem jogará em seu próprio estádio (casa) e quem jogará fora de seu estádio (fora).

O futebol no Brasil é um dos esportes mais importantes do país, além de seu impacto social faz girar também a economia, já que são milhões de pessoas envolvidas com esse esporte, conseguindo até driblar a crise, segundo Capelo (2016). Para isso deve-se criar campeonatos competitivos para que continue atraindo fãs de todo o Brasil.

O Campeonato Brasileiro de Futebol é um dos eventos mais importantes do Brasil. Criado e administrado pela Confederação Brasileira de Futebol a competição atrai diversos fãs que sempre acompanham seus times durante todo andamento. É um campeonato extenso, geralmente iniciado em Maio e terminado 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 o outro. Este trabalho abordará apenas as tabelas geradas para a Série A do campeonato.

O modelo do campeonato brasileiro utiliza duplo *round robin* e tem como critério de classificação os pontos corridos, o qual o time que conseguir mais pontos consagra-se campeão (Confederação Brasileira de Futebol, 2016). Em cada vitória, o time soma 3 pontos, já em caso de empate, ambos os times somam 1 ponto. Da Série A participam 20 times que jogam entre si em *turno e retorno*, sendo 19 jogos no primeiro turno e 19 jogos no segundo turno. Um time só pode jogar uma vez na rodada. Quando os times se enfrentam no primeiro turno, eles voltam a se enfrentar novamente na mesma rodada do segundo turno com mandos de campo invertido. Se ao término do campeonato dois ou mais times possuírem o mesmo número de pontos será considerado como desempate os seguintes critérios: 1º) maior número de vitórias; 2º) maior saldo de gols; 3º) maior número de gols pró; 4º) confronto direto; 5º)

menor número de cartões vermelhos recebidos; 6º) menor número de cartões amarelos recebidos; 7º) sorteio.

## 1.1 Objetivos

O objetivo geral deste projeto 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, com intuito de minimizar o tempo médio percorrido pelas equipes. O problema consiste em alocar os jogos, agrupados em turno e retorno, para as equipes da série em questão.

Os objetivos específicos são:

1. Aplicar um modelo de estudo para o problema, identificando o objetivo a ser otimizado, bem como as restrições a serem observadas para a elaboração da tabela;
2. Estudar métodos para representação do problema, identificando abordagens e estruturas de dados específicas para o *timetabling*;
3. Definir e implementar uma meta-heurística para o problema baseada em técnicas de computação evolucionária;
4. Planejar e realizar experimentos para avaliar a performance da meta-heurística proposta;
5. Analisar e discutir os resultados obtidos, além de identificar possíveis melhorias e considerações gerais sobre o processo.

## 1.2 Estrutura do trabalho

O restante do trabalho está dividido da seguinte forma. O Capítulo 2 apresenta as restrições indicadas pela CBF, além de descrever o porquê da minimização do tempo e o estudo a trabalhos correlatos; A explicação do algoritmo proposto, mostrando como foi implementado e como é representada a solução pode ser visto no Capítulo 3; Os resultados, com a definição do algoritmo, com a definição dos parâmetros utilizados e a comparação dos resultados obtidos, são descritos no Capítulo 4; O Capítulo 5 apresenta as conclusões deste trabalho, além dos possíveis propostas de continuidade.

## 2 Formulação do problema

Este capítulo apresenta a definição do problema a ser estudado, com as considerações acerca dos critérios utilizados para a geração da tabela de jogos. É importante observar que essa definição abordou questões que podem resultar no desgaste das equipes, como o tempo de deslocamento entre as partidas. Inicialmente, serão discutidos as questões acerca dos critérios identificados para a formulação do problema. Em seguida, a função-objetivo definida para o problema será apresentada. Ao final, trabalhos correlatos na literatura serão discutidos, observando a forma com que os autores resolveram esse mesmo problema utilizando métodos diferentes.

### 2.1 Critérios para a construção da tabela

A definição de uma tabela de jogos que seja adequada, atendendo as restrições e não privilegiando nenhuma equipe, é uma tarefa muito difícil. Além disso, conforme o critério adotado para criar uma tabela, indiferente da forma escolhida, equipes podem ser favorecidas.

Quando um time percorre uma distância maior que os outros times sugere-se que ele tenha um desgaste maior, além de mais gastos segundo Agência de Futebol do Interior (2015). Entende-se que isso pode ocasionar num rendimento menor na competição quando comparado aos times que percorreram menores distâncias média. Segundo Dechichi (apud Agência de Futebol do Interior, 2015) para minimizar esse desgaste, seria adequado que a diferença de distância do time que percorreu a menor distância e o time que percorreu a maior distância seja a menor possível.

Entretanto, durante a definição do critério a ser considerado como objetivo a ser otimizado, observou-se que o contexto mais adequado é a minimização do tempo médio de deslocamento. Quanto maior a distância percorrida pelo time, maior é o tempo que irá passar viajando isso ocorre porque os times fazem viagens longas que ocorrem pelo formato de jogos do torneio: jogo em casa / jogo fora de casa.

Outro fator que foi levado em consideração é que a distância de uma cidade para outra não é direta. Como nem todas as cidades que possuem times dispõem de aeroportos, as equipes viajam de avião até uma cidade próxima e terminam a viagem de ônibus. Eles 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. A distância de Belo Horizonte para Santos e Belo Horizonte para São Paulo são parecidas, mas o tempo de viagem são diferentes. Como Santos não possui aeroporto, uma

opção para chegar naquela cidade partindo de Belo Horizonte, é ir de avião para São Paulo e depois ir de ônibus para Santos.

Tabela 1. Distância e tempo de Belo Horizonte até Santos e São Paulo

Origem	Destino	Distância em Linha Reta (km)	Tempo
Belo Horizonte	São Paulo	491,40	1h 15 min
Belo Horizonte	Santos	514,26	2h 45 min

Considerando as questões acerca do deslocamento, a função objetivo foi definida a partir do somatório dos tempos gastos para as equipes irem dos locais atuais para o local da partida. As considerações sobre essa função, bem como as restrições utilizadas são apresentadas a seguir.

## 2.2 Função objetivo

A função objetivo desde trabalho, a qual deve ser minimizada, é representada pela Equação (1).

$$\text{Min } \sum_{i,j,k} T_{ij}L_{ijk} \quad (1)$$

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 representa 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 uma série de regras específicas que devem ser atendidas para que a tabela do campeonato seja considerada viável. Segundo ela, essas restrições têm como objetivo deixar o campeonato mais competitivo.

As restrições consideradas neste trabalho são apresentadas a seguir, e foram retiradas de Ribeiro e Urrutia (2008). Elas são divididas nas seguintes classes:

1. *Restrições do Torneio*: são as restrições básicas de um torneio duplo round robin;
2. *Padrão Casa-Fora*: define de como deve ser o padrão de jogos em casa e times fora de casa;
3. *Clássicos ou Jogos Regionais*: define como serão os clássicos e os jogos que ocorrem dos times de mesmo estado;
4. *Restrições Geográficas*: define como ocorrerá os jogos fora do estado.

A restrição que trata da *correspondência perfeita das equipes emparelhadas* não será considerada neste trabalho, pois parte do princípio que grupos de equipes principais devem ter seus jogos agrupados, sendo que A e B são do mesmo par e o time A jogar em casa contra time C, o time B deve jogar fora contra o time C no mesmo turno.

Além disso, as restrições estabelecidas pela empresa detentora dos direitos de imagem dos jogos que descrevem questões acerca do horário e dia que os jogos devem acontecer, o que não é contemplado neste trabalho. As restrições Geográficas que restringe a quantidade de jogos seguidos nos estados não são consideradas nesse trabalho, pois não foi observado como vantagem jogar mais de 5 jogos no mesmo estado. A restrição que todos os times devem jogar a mesma quantidade de jogos seguidos em casa e fora de casa da classe Padrão Casa-Fora não serão observadas pois nem todos os times vão conseguir ter a mesma quantidade de jogos seguidos em casa e fora de casa durante o torneio. O conjunto completo das restrições, além das demais considerações pertinentes ao contexto, podem ser obtidas no trabalho daquele autor.

### **2.2.1 Restrições do Torneio**

- A1 Cada equipe deve jogar contra outra equipe duas vezes sendo uma em casa e outra fora de casa.
- A2 Cada equipe deve jogar apenas uma vez em cada rodada, seja em casa ou fora (cronograma compacto).
- A3 Cada equipe deve jogar contra outra equipe exatamente uma vez no primeiro e no segundo turno, ao longo do n-1 rodadas. Jogos na segunda fase são reproduzidas exatamente na mesma ordem como na primeira, mas com locais intercambiados. Consequentemente, a programação da segunda fase é diretamente determinada de acordo com a primeira.

### **2.2.2 Padrão Casa-Fora**

- B1 As equipes alternam sua condição de jogo (em casa ou fora) em cada rodada durante as primeiras quatro rodadas da primeira fase e durante as duas últimas rodadas da segunda fase. Em outras palavras, não há jogos seguidos nas quatro primeiras rodadas ou nas duas últimas rodadas do torneio.
- B2 Equipes que jogam em casa na primeira rodada necessariamente jogam fora na segunda, e vice-versa.
- B3 Cada equipe joga 19 jogos por turno sendo que jogará em cada turno um jogo a mais em casa ou fora.

B4 Cada equipe só pode jogar duas vezes em casa-fora em rodadas consecutivas.

### 2.2.3 Clássicos ou Jogos Regionais

C1 Nenhum jogo “clássico” pode ser jogado nas três primeiras rodadas (no início da competição eles não chamam as grandes audiências, porque os fãs estão menos motivados) ou jogos regionais/clássicos nas últimas quatro rodadas (Para evitar as programações de viagens que dão e vantagem ou desvantagem para uma equipe em uma cidade que hospeda várias equipes) do torneio.

C2 Não pode haver mais que um jogo “clássico” jogado na mesma cidade em qualquer rodada. Essa restrição é imposta por conta da logística e segurança.

C3 Nenhuma equipe pode jogar dois jogos clássicos em rodadas consecutivas, uma vez que eles são geralmente jogos mais difíceis para os quais as equipes solicitam mais tempo entre jogos. Perder dois jogos clássicos seguidos afetaria fortemente a motivação de uma equipe e do interesse dos seus fãs.

### 2.2.4 Restrições Geográficas

D1 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.

A seção a seguir apresentará trabalhos correlatos na literatura. Em cada um deles será apresentada a técnica utilizada, bem como os critérios e as restrições utilizados na abordagem do problema.

## 2.3 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 era a minimização dos gastos financeiros com o deslocamento. Resultados relevantes foram apresentados, minimizando a distância média percorrida entre os times quando comparada com a tabela oficial.

Bartsch, Drexler e Kröger (2004) utilizaram de 3 heurísticas para criar um calendário que cumpria os requisitos estabelecidos para as ligas Alemã e Australiana, e criar um calendário utilizável para as ligas. As heurísticas desenvolvidas foram de *Semi-greedy*, *Truncated branch-and-bound* e *Exact branch-and-bound* para criar um calendário para liga alemã e utilizou a heurística de *Semi-greedy* para criar um calendário para a liga Australiana. Seu maior êxito foi conseguir criar alguns calendários em um tempo pequeno.

Rasmussen e Trick (2006) utilizaram de Programação Inteira (PI) e criação de um modelo de restrições (CMR) além de um algoritmo *branch and price* para resolver o problema de minimização de distância de jogos para torneios *round robin*. A utilização do algoritmo híbrido de PI/CMR mostrou bons resultados e foi definido pelo autor como melhor método utilizado, já o algoritmo de *Branch and Price* não teve resultados computacionais tão bons quanto o PI/CMR.

Ribeiro e Urrutia (2008) abordaram o problema por meio de Programação Linear Inteira. Na ocasião, 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. Nesse ano, o campeonato considerado foi o mais atraente, pois o título foi disputado até a última rodada por quatro times.

## **2.4 Considerações Finais**

Este capítulo apresentou o problema dos desgastes que acontecem quando se faz muitas viagens e viagens longas. Também foram apresentadas as restrições que serão adotadas neste trabalho, além de abordagens correlatas na literatura. No capítulo seguinte será descrito o algoritmo de maneira geral, apresentado seu pseudocódigo, e explicando o funcionamento de cada etapa do processo. Será apresentada também a maneira utilizada na representação das soluções, bem como os operadores utilizados e a geração da população inicial.

### 3 Algoritmo Proposto

O algoritmo proposto é baseado na meta-heurística Estratégia Evolutiva ( $\mu + \lambda$ ) (EE, do inglês *Evolution Strategy*), a qual é uma classe de algoritmos evolutivos. De acordo com Luke (2013), os passos principais para criação da EE podem ser definidos como segue. O processo é iniciado com um número  $\mu$  de indivíduos gerados aleatoriamente. As ações seguintes definem o processo de iteração sobre a população. A *fitness*, é uma função de avaliação de todos os indivíduos. O  $\lambda$  é a quantidade de descendentes que será criado a partir dos  $\mu$  indivíduos. Cada indivíduo irá gerar  $\lambda/\mu$  descendentes por meio de mutação. Após as mutações, os descendentes irão competir com os pais por meio de seus *fitness*. Para a próxima geração selecionados os  $\mu$  indivíduos mais aptos do conjunto formado por  $(\mu + \lambda)$  indivíduos.

Esse algoritmo faz a reparação de algumas restrições violadas durante a execução da mutação e terá como objetivo minimizar o tempo de viagem dos times do campeonato brasileiro. No tópico seguinte será explicado o funcionamento do algoritmo desenvolvido.

#### 3.1 Pseudocódigo do algoritmo proposto

O Algoritmo 1.1 apresenta de modo detalhado o algoritmo desenvolvido para solucionar o problema abordado nesse trabalho.

O EE foi a estratégia escolhida por utilizar como operador principal a **mutação**. O processo seria mais complexo se fosse utilizada algum método que emprega o *crossover*, pois a definição do cruzamento poderia implicar na violação de diversas restrições. Os operadores descritos na Seção 3.4 serão utilizados na mutação. Os passos de execução do algoritmo 1.1 são descritos a seguir.

A execução do EE do algoritmo recebe como entrada o tamanho de  $\mu$  e o tamanho de  $\lambda$  e define o tamanho da sua condição de parada que é o máximo de evoluções estipulados. Em seguida define o tamanho da população dos melhores indivíduos que é  $\mu$  (linha 2) e da população dos descendentes mais os melhores indivíduos que é  $\mu + \lambda$  (linha 3). Depois lê os operadores de mutação que serão usados no programa e inicia a criação da população inicial de tamanho  $\mu$  (linha 4 até 9). O processo de criação é definido na Seção 3.3. A linha 8 representa o incremento de evolução para cada indivíduo criado. Quando terminar de criar a população inicial, ela vai ser ordenada de acordo com seu melhor resultado (linha 10).

Após a criação da população inicial o algoritmo irá gerar os descendentes. Para cada indivíduo em  $\mu$  é gerado  $\lambda/\mu$  descendentes (linha 12). Esses descendentes são gerados a

**Algoritmo 1.1**

```

EE( $\mu$ ,  $\lambda$ , MaxEvoluções)
1: evolucoes <- 0
2: P <- {} População de  $\mu$  indivíduos
3: P' <- {} População  $\mu + \lambda$ 
4: para  $\mu$  vezes faça
5:   P <- P U (Novos indivíduos aleatórios)
6:   CalculaFitness (P $\mu$ )
7:   AvaliaFitness (P $\mu$ )
8:   evolucoes <- evolucoes + 1
9: fim-para
10: ordena(P $\mu$ )
11: enquanto (evolucoes < MaxEvoluções) faça
12:   para P $\mu$   $\in$  P vezes faça
13:     para  $\lambda/\mu$  vezes faça
14:       D = P $\mu$  > Cria Descendentes
15:       executaoperadores(D)
16:       CalculaFitness (D)
17:       AvaliaFitness (D)
18:       P' = P' U (D)
19:       evoluções <- evoluções + 1
20:     fim-para
21:   fim-para
22:   para P $\mu$   $\in$  P vezes faça
23:     P' = P' U (P $\mu$ )
24:   fim-para
25:   limpa(P)
26:   ordena(P')
27:   para  $\mu$  vezes faça
28:     P = P U (P' $\mu$ )
29:   fim-para
30:   limpa(P')
31: fim-enquanto
32: retorna P
Fim EE

```

partir da execução dos operadores de mutação (linha 15), descritos na Seção 3.4. A linha 19 representa o incremento do número de gerações. Após a execução dos operadores, o *fitness* do descendente é calculado que é o tempo gasto de viagem dos times durante o campeonato (linha 16), depois é avaliado se algum time está violando alguma restrição (linha 17), se houver violação o indivíduo é punido acrescentando um tempo maior no seu fitness de acordo com a quantidade de vezes que a restrição foi violada e ele é adicionado na população total de descendentes.

Assim que todos os  $\mu$  indivíduos criarem seus descendentes, faz-se a união dos descendentes com os  $\mu$  indivíduos (linha 22 até 24) e realiza a comparação de todos eles

(linha 26), salvando os  $\mu$  melhores indivíduos (linha 27 até 29) e descartando os outros. Executa-se o passo de criação de descendentes até que a condição de parada seja satisfeita. Quando satisfeita, retorna a população que contém os  $\mu$  melhores resultados (linha 32).

### 3.2 Representação das soluções

Cada indivíduo da população será representado por uma matriz, afim de demonstrar a tabela e um vetor contendo N posições no qual são armazenadas as informações dos times. Assim, a representação completa de cada indivíduo é uma matriz com  $2N-2$  linhas (Rodadas) e  $N/2$  colunas (Jogos). Todos operadores serão aplicados no primeiro turno do campeonato. O segundo turno consiste no primeiro turno com os mandos de campo invertidos.

O campeonato brasileiro atual consiste em 38 rodadas que são divididas em dois turnos de 19 rodadas. Cada uma dessas rodadas possui 10 jogos (20 times), e cada célula da matriz<sub>ij</sub> representa um jogo entre os times. Por exemplo, no formato do campeonato atual com 20 times, um fragmento de uma tabela resultado é ilustrado na Figura 1.

Figura 1. Representação de uma tabela

VETOR DE TIMES																							
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 DESTACANDO AS 3 PRIMEIRAS RODADAS E 3 ÚLTIMAS										REPRESENTAÇÃO DOS TIMES NO VETOR													
AxB	CxD	ExF	GxH	IxJ	KxL	MxN	OxP	QxR	SxT	A -> FLAMENGO	K -> BOTAFOGO	BxC	RxI	HxM	DxE	FxG	TxQ	JxO	PxA	LxS	BxK	B -> SPORT	L -> SÃO PAULO
KxD	MxF	QxJ	AxT	CxR	IxH	GxL	OxN	SxB	ExP	C -> PALMEIRAS	M -> FIGUEIRENSE	⋮											
BxD	MxJ	ExI	LxT	RxN	GxA	FxH	SxO	KxC	QXP	E -> ATLETICOPR	O -> CORINTHIANS	JxG	BxQ	FxA	DxO	NxK	PxI	LxE	HxS	TxC	RxM	F -> SANTOS	P -> GREMIO
AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP	G -> CORITIBA	Q -> AMERICAMG	AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP	H -> CRUZEIRO	R -> FLUMINENSE
										I -> SANTA CRUZ	S -> INTERNACIONAL											J -> VITORIA	T -> CHAPECOENSE

Com esta representação é possível observar que em um campeonato com 20 times serão disputadas 38 rodadas, e que cada rodada possui 10 jogos. Os turnos são espelhados, e todos times jogam em todas as rodadas.

O vetor de 20 posições contém todas informações dos times participantes do campeonato brasileiro atual. Eles são representados na tabela pelo seu índice no vetor mais 1, ou seja, se o Cruzeiro está no índice 7 ele será representado na tabela pelo valor 8. Os jogos são as combinações desses números separados. Para efeito de ilustração, os confrontos são separados por um "X". O time que estiver ao lado esquerdo do é o será o time que jogará em seu estádio, ou seja, o mandante do jogo (casa) e o time que estiver ao lado direito será o visitante (fora).

Um time que está armazenado no vetor contém um conjunto de informações sendo elas (i) identificação (ii) Nome do Time (iii) Cidade (iv) Estado (v) Onde está localizado (vi) Tempo de viagem gasto (vii) Identificação de Clássicos. Essas informações são explicadas a seguir.

A identificação representa o time na matriz resultado, a localização mostra em qual cidade o time está em uma determinada rodada, o tempo mostra o tempo gasto no percurso de toda a tabela do campeonato. Os clássicos são marcadores que denotam com quais times há uma disputa de clássico.

Por exemplo, o time do Palmeiras é identificado por 2, sua cidade de origem é São Paulo, seu estado é São Paulo e Palmeiras e Santos fazem disputa de clássico pois possuem o mesmo marcador 2. A Figura 2 mostra detalhadamente a estrutura que o vetor utiliza antes de iniciar o campeonato.

Figura 2. Exemplo da estrutura do vetor de times antes do início dos jogos.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
ID	NOME	CIDADE	ESTADO	LOCALIZAÇÃO	TEMPO	CLASSICO													
A	FLAMENGO	RIO DE JANEIRO	RIO DE JANEIRO	RIO DE JANEIRO	0	0													
B	SPORT	RECIFE	PERNAMBUCO	RECIFE	0	1													
C	PALMEIRAS	SÃO PAULO	SÃO PAULO	SÃO PAULO	0	2													
D	ATLETICOPR	CURITIBA	PARANA	CURITIBA	0	3													
E	ATLETICOMG	BELO HORIZONTE	MINAS GERAIS	BELO HORIZONTE	0	4													
F	SANTOS	SANTOS	SÃO PAULO	SANTOS	0	2													

A Figura 3 mostra como fica o vetor na segunda rodada atualizando onde os times estão no Brasil e o tempo de viagem percorrido até o momento.

Figura 3. Exemplo da estrutura do vetor de times na rodada 2

REPRESENTAÇÃO DA TABELA DE JOGOS DESTACANDO AS 3 PRIMEIRAS RODADAS E 3 ÚLTIMAS									
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

ID	NOME	CIDADE	ESTADO	LOCALIZAÇÃO	TEMPO EM MINUTOS	CLASSICO
A	FLAMENGO	RIO DE JANEIRO	RIO DE JANEIRO	PORTO ALEGRE	98	0
B	SPORT	RECIFE	PERNAMBUCO	RECIFE	200	1
C	PALMEIRAS	SÃO PAULO	SÃO PAULO	CAMPINAS	95	2
D	ATLETICOPR	CURITIBA	PARANA	CURITIBA	120	3
E	ATLETICOMG	BELO HORIZONTE	MINAS GERAIS	CURITIBA	105	4
F	SANTOS	SANTOS	SÃO PAULO	SANTOS	330	2

Exemplo da estrutura do vetor de times na rodada 2

**3.3 Geração da População Inicial**

Uma tabela do campeonato brasileiro, como o exemplo apresentado na Figura 1, será utilizada como modelo para ilustrar a geração da população inicial com  $\mu$  de indivíduos. Os times serão embaralhados, tendo seus jogos alterados aleatoriamente e o modelo indicará os confrontos como na Figura 4. Assim, a estrutura da tabela não é alterada. Após essa operação, os indivíduos serão avaliados e ordenados de acordo com seu *fitness*.

Figura 4. Tabela com times misturados aleatoriamente.

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 DESTACANDO AS 3 PRIMEIRAS RODADAS E 3 ÚLTIMAS									
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

REPRESENTAÇÃO DOS TIMES NO VETOR	
D -> FLAMENGO	I -> BOTAFOGO
L -> SPORT	N -> SÃO PAULO
F -> PALMEIRAS	A -> FIGUEIRENSE
E -> ATLETICOPR	I -> PONTE PRETA
S -> ATLETICOMG	K -> CORINTHIANS
T -> SANTOS	C -> GREMIO
M -> CORITIBA	Q -> AMERICAMG
F -> CRUZEIRO	R -> FLUMINENSE
P -> SANTA CRUZ	O -> INTERNACIONAL
G -> VITORIA	B -> CHAPECOENSE

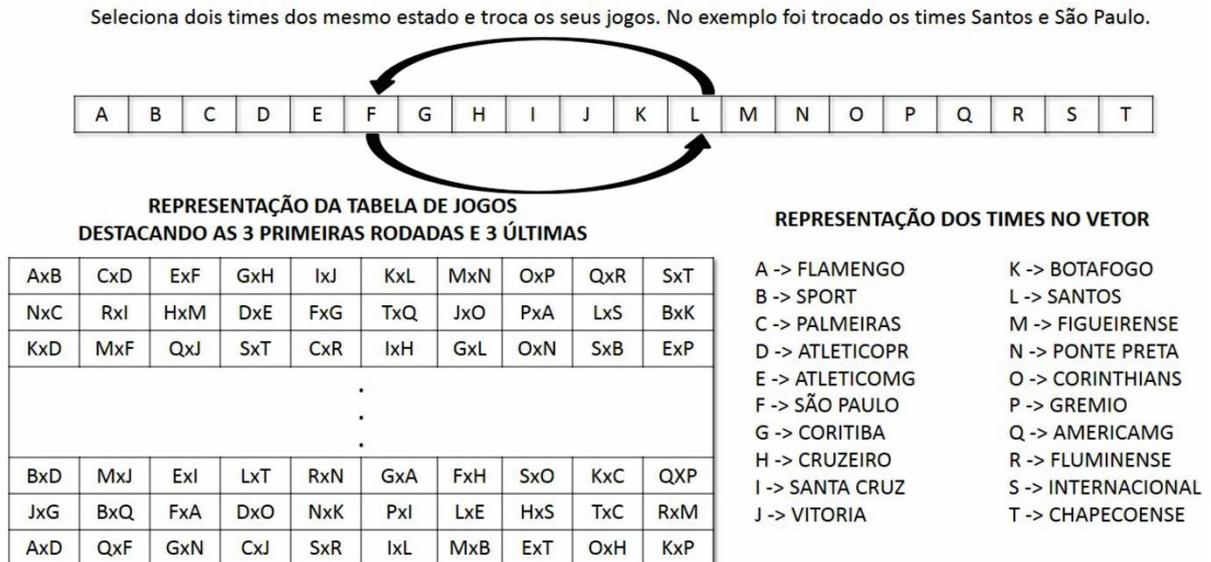
### 3.4 Operadores utilizados

O algoritmo irá utilizar 4 tipos de operadores de mutação, sendo possível selecionar as combinações entre eles. Os operadores são: *Troca de times regionais*; *Troca de times com reparo de clássicos nas 3 primeiras rodadas utilizando times de estados diferentes*; *Troca de times com reparo de clássicos nas 3 primeiras rodadas mudando times entre os clássicos*; e a *Troca de rodadas com reparo de jogos seguidos mudando mando de campo*. Esses operadores serão aplicados no primeiro turno e espelhado para o segundo turno. A seguir, cada um dos operadores será descrito.

#### 3.4.1 Troca de times regionais

Nesse operador, os novos indivíduos serão criados a partir da troca de times do mesmo estado. Esse processo consiste em selecionar dois times que mandam seus jogos no mesmo estado e trocar suas partidas umas pelas outras. Ou seja, um time F é trocado pelo time L sendo que F e L são do mesmo estado. Então, esse time F assumirá as partidas do time L, e o time L assumirá as partidas do time F. Essa operação é apresentada na Figura 5.

Figura 5. Troca de times regionais



Após essa operação, os times serão avaliados e ordenados de acordo com a sua *fitness*. Fazendo isso, a estrutura da tabela não é alterada, mas o tempo médio dos times trocados foram alterados já que a forma com que eles vão viajar foi modificada. Esse operador não foi utilizado nos resultados pois nos experimentos feitos o ganho de tempo é pequeno ou zero. Isso ocorre porque não existe muitos times do mesmo estado que não são clássicos ou não são de diferentes cidades.

### 3.4.2 Troca de times com reparo de clássicos nas 3 primeiras rodadas utilizando times de estados diferentes

Esse operador é executado em duas etapas. Na primeira, os novos indivíduos serão criados a partir da troca de times. Isso é feito simplesmente selecionando dois times aleatórios definido pelo algoritmo e trocando suas partidas umas pelas outras, ou seja, as partidas de um time A é trocada pelas partidas de um time B, então esse time A assumirá as partidas do time B e o time B assumirá as partidas do time A como na Figura 6.

Figura 6. Troca de times aleatórios



Após essa operação, os times serão avaliados para verificar se contém clássicos nas 3 primeiras rodadas como mostra a Figura 7.

Figura 7. Troca de times aleatórios



Se houver, um dos times que disputa o clássico será selecionado aleatoriamente e seus jogos serão trocados por um time de outro estado também selecionado aleatoriamente como mostrado na Figura 8.

Figura 8. Troca um time dos clássicos por um time aleatório de outro estado



Será feita a etapa de verificação e conserto até que o indivíduo não contenha mais clássicos nas 3 primeiras rodadas. Quando não conter mais clássicos ele será avaliado e ordenado de acordo com a sua fitness.

Fazendo isso, a estrutura da tabela não é alterada, mas o tempo total dos times trocados foram alterados já que a forma com que eles vão viajar foi modificada.

### 3.4.3 Troca de times com reparo de clássicos nas 3 primeiras rodadas mudando times entre os clássicos

Esse operador é executado em duas etapas sendo a primeira os novos indivíduos serão criados a partir da troca de times, que é simplesmente selecionar dois times aleatórios definido pelo algoritmo e trocar suas partidas umas pelas outras.

Um exemplo, o time A foi selecionado aleatoriamente e seus jogos são trocados pelos jogos do time B que também é selecionado aleatoriamente, como mostra a Figura 9.

Figura 9. Troca de times aleatórios



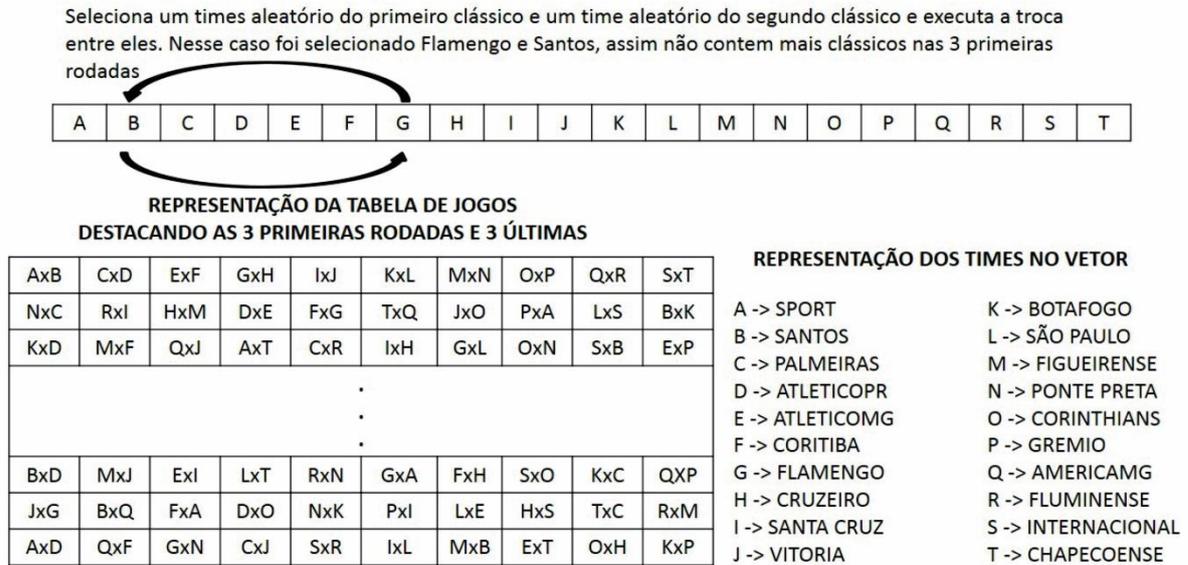
Após essa operação os times serão avaliados para verificar se contém clássicos nas 3 primeiras rodadas assim como na Figura 10.

Figura 10. Verifica se há clássicos nas 3 primeiras rodadas



Se houverem dois ou mais clássicos que os times correspondentes não são clássicos entre si, o algoritmo executará a segunda etapa que é selecionar um dos times que disputa o primeiro clássico e outro time que disputa o segundo clássico, e seus jogos serão trocados assim como na Figura 11.

Figura 11. Troca times entre clássicos da primeira rodada



Se o procedimento for executado durante 5 vezes sem nenhuma melhora (nenhum clássico), ou encontrar apenas um clássico, ou todos os clássicos de times que também são clássicos entre si, o algoritmo passa para próxima etapa.

Neste caso, um dos times que disputa o clássico será selecionado aleatoriamente e seus jogos serão trocados por um time de outro estado também selecionado aleatoriamente. Após isso o indivíduo será avaliado novamente. Caso ainda exista clássicos nas 3 primeiras rodadas, será executado novamente a etapa que conserta os clássicos, fazendo isso até não existir mais clássicos nas 3 primeiras rodadas. No final, ele será avaliado e ordenado de acordo com a sua fitness.

Fazendo isso, a estrutura da tabela não é alterada, mas o tempo médio dos times trocados foram alterados já que a forma com que eles vão viajar foi modificada.

### 3.4.4 Troca de rodadas com reparo de jogos seguidos mudando mando de campo

Esse operador é executado em duas etapas, sendo que a primeira os novos indivíduos serão criados a partir da troca de rodadas. Serão escolhidas aleatoriamente duas rodadas entre as 5 e 17 e trocados seus jogos. Por exemplo, se selecionada a rodada 7 e 13, os jogos da rodada 7 irá acontecer na rodada 13 e os da rodada 13 irá acontecer na rodada 7 como na Figura 12.

Figura 12. Seleciona duas rodadas aleatórias e troca seus jogos. No exemplo foi selecionado para trocar as rodadas 7 e 13.

Rodada	Jogos									
...	...									
4	AxB	CxD	ExF	GxH	IxJ	KxL	MxN	OxP	QxR	SxT
5	NxC	RxI	HxM	DxE	FxG	TxQ	JxO	PxA	LxS	BxK
6	KxD	MxF	QxJ	AxT	CxR	IxH	GxL	OxN	SxB	ExP
7	FxK	CxM	BxI	LxA	TxO	DxH	SxQ	RxG	JxE	NxP
8	TxD	KxM	FxJ	IxS	AxR	QxC	OxB	HxN	PxL	GxE
9	JxK	MxT	DxF	HxR	IxA	NxQ	CxP	OxL	GxS	ExB
10	BxD	MxJ	ExI	LxT	RxN	AxG	FxH	SxO	KxC	QXP
11	JxG	BxQ	FxA	DxO	NxK	PxI	LxE	HxS	TxC	RxM
12	AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP
13	KxT	DxM	FxI	JxA	RxO	CxH	LxQ	BxG	NxE	SxP
...	...									

Após essa troca de rodada, é verificado se algum time joga mais de dois jogos seguidos em casa ou fora de casa, como ilustrado pela Figura 13. Caso haja algum time que infrinja essa regra, verifica se o erro começou com a rodada 4. Se for verdade seleciona aleatoriamente a rodada 5 ou 6, e inverte o mando de campo do jogo do time que infringiu a regra. A rodada 4 não pode ser alterada pelo fato de que as quatro primeiras rodadas os times devem jogar com o mando de campo invertido.

Figura 13. Verifica se há algum time com mais de dois jogos seguidos em casa ou fora de casa. Nesse caso o time K tem 3 jogos seguidos em casa.

Rodada	Jogos									
...	...									
4	AxB	CxD	ExF	GxH	IxJ	KxL	MxN	OxP	QxR	SxT
5	NxC	RxI	HxM	DxE	FxG	TxQ	JxO	PxA	LxS	BxK
6	KxD	MxF	QxJ	AxT	CxR	IxH	GxL	OxN	SxB	ExP
7	KxT	DxM	FxI	JxA	RxO	CxH	LxQ	BxG	NxE	SxP
8	TxD	KxM	FxJ	IxS	AxR	QxC	OxB	HxN	PxL	GxE
9	JxK	MxT	DxF	HxR	IxA	NxQ	CxP	OxL	GxS	ExB
10	BxD	MxJ	ExI	LxT	RxN	AxG	FxH	SxO	KxC	QXP
11	JxG	BxQ	FxA	DxO	NxK	PxI	LxE	HxS	TxC	RxM
12	AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP
13	FxK	CxM	BxI	LxA	TxO	DxH	SxQ	RxG	JxE	NxP
...	...									

Se o erro não aconteceu com a rodada 4, uma das 3 rodadas é escolhida aleatoriamente. O mando de campo do jogo do time que infringiu a regra é invertido, como representado na Figura 14. Após a troca de mando de campo, verifica se há mais times com jogos seguidos. É feito isso até que não haja nenhum jogo seguido na tabela.

Figura 14. Seleciona aleatoriamente um dos jogos. No caso foi selecionado o jogo KxT e invertido seus mandos de campo.

Rodada	Jogos									
...	...									
4	AxB	CxD	ExF	GxH	IxJ	KxL	MxN	OxP	QxR	SxT
5	NxC	RxI	HxM	DxE	FxG	TxQ	JxO	PxA	LxS	BxK
6	KxD	MxF	QxJ	AxT	CxR	IxH	GxL	OxN	SxB	ExP
7	<b>TxK</b>	<b>DxM</b>	FxI	JxA	RxO	CxH	LxQ	BxG	NxE	SxP
8	TxD	KxM	FxJ	IxS	AxR	QxC	OxB	HxN	PxL	GxE
9	JxK	MxT	DxF	HxR	IxA	NxQ	CxP	OxL	GxS	ExB
10	BxD	MxJ	ExI	LxT	RxN	AxG	FxH	SxO	KxC	QXP
11	JxG	BxQ	FxA	DxO	NxK	PxI	LxE	HxS	TxC	RxM
12	AxD	QxF	GxN	CxJ	SxR	IxL	MxB	ExT	OxH	KxP
13	FxK	CxM	BxI	LxA	TxO	DxH	SxQ	RxG	JxE	NxP
...	...									

### 3.5 Considerações Finais

Neste capítulo foi apresentado o algoritmo proposto, com as devidas considerações em cada etapa. Os operadores definidos serão utilizados no processo de mutação com o propósito de criar novas tabelas tentando não mudar a estrutura da tabela e diminuir o número de restrições violadas. A proposta de criação da população inicial a partir de um modelo tem como objetivo também gerar soluções com o número mínimo possível de violações.

No próximo capítulo será discutido a implementação do algoritmo proposto, o ambiente computacional utilizado para a realização dos experimentos, bem como os parâmetros utilizados. Além disso, os resultados serão comparados com as tabelas oficiais do campeonato brasileiro, apresentando as devidas considerações e discussões.

## 4 Resultados

Este capítulo apresenta, inicialmente, a implementação do algoritmo, a linguagem utilizada e demais considerações. Em seguida, serão apresentados os parâmetros utilizados e as questões acerca da sua definição. Os resultados são relatados e discutidos, com a devida análise e considerações.

### 4.1 Implementação do algoritmo

O algoritmo foi desenvolvido na linguagem de programação Java utilizando o *framework* jMetal (*Metaheuristic Algorithms in Java*). Segundo Durillo e Nebro (2011) o *framework* possui código-livre e licença GNU (*Lesser General Public License*). 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.

### 4.2 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 2.00GHz 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  $\mu$  e  $\lambda$  foi possível 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 de gerações foi utilizado como critério de parada para o algoritmo.

Tabela 2. Parâmetros

Tabela de parâmetros	
Gerações	20.000
$\mu$	40
$\lambda$	$3\mu = 120$

### 4.3 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.

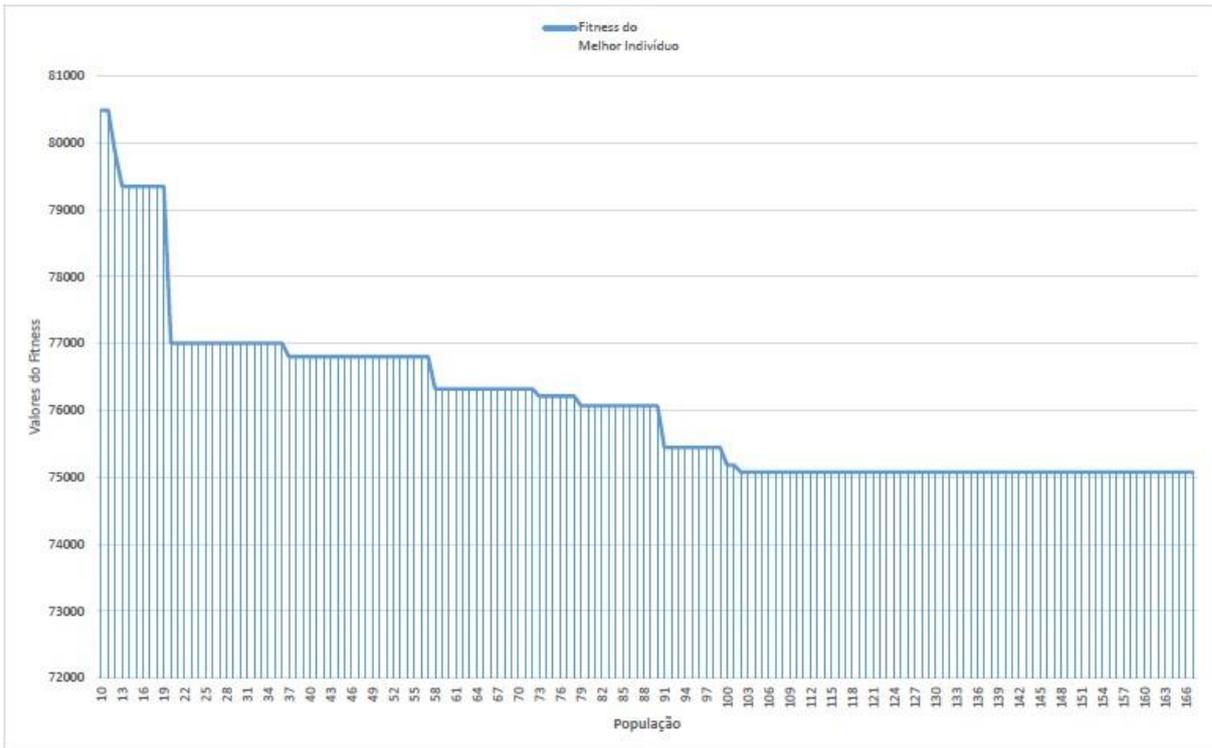
Como instâncias de teste foram utilizadas as seguintes tabelas: Campeonato Brasileiro 2016 (CB2016), Campeonato Brasileiro 2015 (CB2015) e Campeonato Brasileiro 2014 (CB2014), sendo todos da série A. As combinações dos operadores de mutação que foram empregados são os seguintes:

- CO1. Operador troca rodada e mando de campo (Seção 3.4.4) juntamente com operador troca de times com reparo de clássicos nas 3 primeiras rodadas utilizando times de estados diferentes (Seção 3.4.2).
- CO2. Operador troca rodada e mando juntamente com operador troca de times com reparo de clássico nas 3 primeiras rodadas mudando times entre os clássicos (Seção 3.4.3).

Foram realizadas 33 execuções para cada combinação de operadores. O tempo médio de execução para cada uma das execuções foi de 4 minutos.

Considerando o ambiente experimental, os resultados sugerem que o algoritmo é capaz de convergir de maneira satisfatória. A Figura 15 ilustra esse cenário, considerando uma das execuções do algoritmo. Este exemplo apresenta que o melhor indivíduo encontrado de cada população criada é atualizado durante a execução do algoritmo. Para facilitar a visualização, foi considerada apenas a parte em que o indivíduo se torna factível. Anterior a esse ponto, caso o indivíduo violasse as restrições, penalizações são aplicadas aumentando o seu fitness em um valor.

Figura 15. Gráfico de convergência do algoritmo a partir de um valor factível.



Na Tabela 3 é apresentado o resultado das execuções para cada uma das combinações utilizadas. Na tabela foram inseridos o *menor indivíduo* (Menor), o qual corresponde ao melhor valor encontrado pelo algoritmo; o *maior indivíduo* (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

Instância	CO1				CO2			
	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	86.587,55	321,97

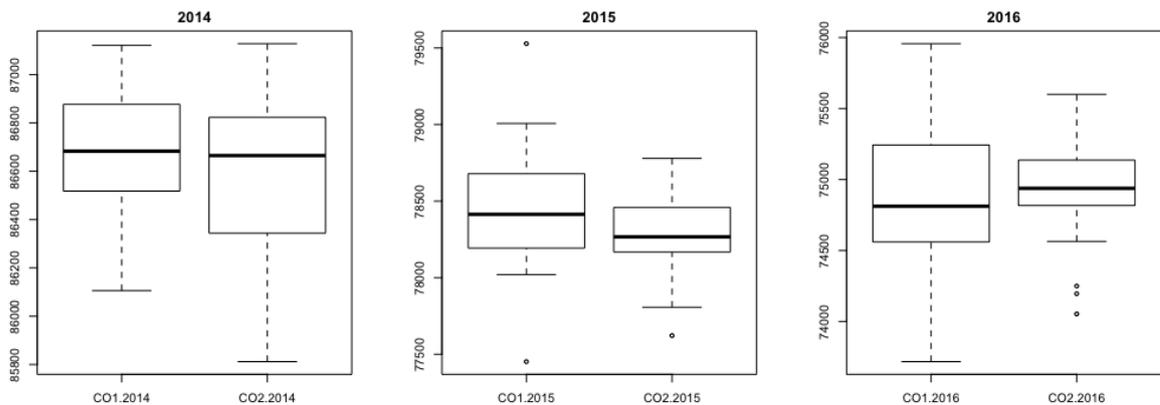
Para avaliar a diferença entre os resultados foi utilizado o cálculo do GAP, definido pela Equação 2, o qual representa o desempenho entre os operados.

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

Os resultados obtidos pelas execuções sugerem que, para a instância CB2016, o operador CO1 foi 0,45% melhor que a CO2. Para a instância CB2015, o operador CO1 foi

0,44% melhor e na instância CB2014, o operador CO2 foi 0,34% melhor. Mesmo os resultados sendo aparentemente próximos, a diferença de tempo entre eles pode ser significativa, considerando que isso pode resultar num maior desgaste pelas equipes. A diferença entre os resultados também pode ser observada pelos *boxplots* apresentados na Figura 16. É possível observar que o valor máximo está mais afastado dos valores centrais, e que utilizando o CO2 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 CO1.

Figura 16: *Boxplots* dos resultados



Com os resultados apresentados pelo algoritmo na Tabela 3, é possível também realizar a comparação do melhor resultado apresentado com os valores obtidos a partir tabela oficial do campeonato e seu respectivo ano, utilizando as restrições observadas e com objetivo de minimizar o tempo total de viagens dos times. Essa comparação é apresentada na Tabela 4 em que é mostrado o menor resultado (*Menor*), a diferença do tempo gasto do time (*DTGT*) que mais passou tempo viajando com o que passou menos, pelo operador 1 e 2 (CO1 e CO2). Para a tabela oficial divulgada pela CBF é apresentado a *fitness*, conforme o cálculo realizado pela Equação 1, e as restrições violadas, conforme o contexto determinado por este trabalho.

Os valores destacados em negrito representam os melhores resultados 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. Comparação dos resultados obtidos com a tabela oficial

Operador	CO1		CO2		Tabela Oficial	
	Menor	DTGT	Menor	DTGT	Fitness	DTGT
CB2016	<b>73.717</b>	<b>2.836</b>	74.053	2.877	80.991	3.631
CB2015	<b>77.452</b>	3.891	77.622	<b>3.613</b>	81.096	3.652
CB2014	86.106	3.998	<b>85.812</b>	<b>3.742</b>	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. A tabela oficial da CBF apresenta que, pelo menos, uma das restrições não foi respeitada de acordo com o contexto observado. Isso pode ter acontecido em virtude de modificações realizadas para atender condições especiais, como possíveis questões impostas empresa detentora dos direitos de imagem dos jogos. Entretanto, tais situações estão fora do escopo deste trabalho.

No Campeonato de 2016, a diferença do total de tempo do melhor resultado da viagem da tabela oficial é de 7.274 minutos em comparação a tabela oficial, além do fato da tabela oficial de 2016 violar a restrição D1. A restrição D1 se refere que cada equipe deve jogar fora de seu estado em uma das duas primeiras rodadas.

Já no campeonato de 2015, a diferença do total de tempo da viagem do melhor resultado com a tabela oficial é de 3.184, e a tabela oficial de 2015 também viola a restrição D1.

O campeonato de 2014 teve a diferença do total de tempo do melhor resultado da viagem com a tabela oficial é igual 3.720, mas a tabela oficial de 2014 violou várias vezes a restrição B1 e uma vez a restrição C1. Tais restrições se referem que nas primeiras quatro rodadas os jogos devem ser alternados (casa e fora) e também não podem ocorrer clássicos nas 3 primeiras rodadas e nem regionais ou clássicos nas 4 últimas.

De acordo com as condições experimentais e o contexto deste trabalho, na maioria dos testes realizados o algoritmo conseguiu gerar resultados com tabelas viáveis e que atendessem todas restrições observadas. Além disso, o algoritmo foi capaz de obter melhora no tempo de viagem total e na diferença do que mais viajou para o que menos viajou. Isso sugere que o algoritmo, mesmo partindo de uma solução inicial embaralhada, consegue chegar em uma solução factível diferente da tabela oficial utilizada pela CBF, por meio das combinações de operadores.

#### **4.1 Considerações Finais**

Conforme as condições experimentais estabelecidas no contexto deste trabalho, considerando também as definições do modelo para o problema, o algoritmo proposto foi capaz de gerar boas soluções. Além disso, os resultados sugerem que os tempos, tanto o total quanto a diferença entre as equipes que menos e mais viajam, foram reduzidos em relação à Tabela Oficial da CBF. As restrições violadas pela tabela oficial podem sugerir fatores externos ao contexto do problema, além de também não serem contemplados no escopo deste trabalho.

O próximo capítulo apresentará as considerações finais acerca deste projeto. Além disso, propostas de continuidade e de trabalhos futuros serão discutidas.

## 5 Considerações Finais

Este trabalho teve como objetivo propor um algoritmo baseado em uma meta-heurística para diminuir o tempo de viagem que os times fazem ao decorrer do campeonato brasileiro. Como o campeonato é longo, a redução desse tempo é importante, pois pode ajudar a reduzir o desgaste das equipes.

O algoritmo proposto utilizou a meta-heurística Estratégia Evolutiva e operadores de troca para modificar a solução corrente. Considerando o contexto experimental, os resultados sugerem que o algoritmo alcançou seus objetivos utilizando qualquer uma das duas combinações de operadores. A redução no tempo de viagem pode ser observada nas tabelas do Campeonato Brasileiro de 2014, 2015 e 2016, além de diminuir o 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 de 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, tornando essa quantidade igual.

Outra ação é o estudo e a inclusão de métodos de busca local eficientes para melhorar os resultados obtidos pelo algoritmo 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 o campeonato seja mais justo e dentro de todas as restrições que são importantes para um calendário competitivo.

## 6 Referências

Agência Futebol Interior (2015). Sport viaja 3,6 vezes mais que trio de ferro e dá quase duas voltas ao mundo no brasileiro. Agência Futebol Interior, 2015

<<https://www.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 abr. 2016.

BARTSCH, Thomas; DREXL, Andreas; KRÖGER, Stefan. Scheduling the professional soccer leagues of Austria and Germany. **Elsivier**. Alemanha, p. 1-30. 13 nov. 2004.

Disponível em: <[http://ac.els-cdn.com/S0305054804002606/1-s2.0-S0305054804002606-main.pdf?\\_tid=cd81f744-0d7e-11e7-88fa-](http://ac.els-cdn.com/S0305054804002606/1-s2.0-S0305054804002606-main.pdf?_tid=cd81f744-0d7e-11e7-88fa-00000aab0f02&acdnat=1490022572_6b8b1c503333425c88479dd5062a625c)

00000aab0f02&acdnat=1490022572\_6b8b1c503333425c88479dd5062a625c>. Acesso em: 18 abr. 2017.

BLAJOLI, Fabrício L.. Resolução do Problema de Programação de Jogos do Campeonato Brasileiro de Futebol. 2003. Departamento de Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto.

CAPELO, Rodrigo. Futebol brasileiro volta a crescer em 2015, e primeira divisão fatura R\$ 3,6 bilhões. *Época*, 2016. Disponível em:

<<http://epoca.globo.com/vida/esporte/noticia/2016/05/futebol-brasileiro-volta-crescer-em-2015-e-primeira-divisao-fatura-r-35-bilhoes.html>>. Acesso em: 10 de mar. de 2017.

CONFEDERAÇÃO BRASILEIRA DE FUTEBOL específico da competição campeonato brasileiro da série a 2016. Confederação Brasileira de Futebol, 2016

<[http://cdn.cbf.com.br/content/201604/20160406101909\\_0.pdf](http://cdn.cbf.com.br/content/201604/20160406101909_0.pdf)>. Acessado em: 20 de abr. 2016.

DURILLO, J. J. e NEBRO, A. J.. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 2011, 42(10):760–771.

LUKE, S. *Essentials of Metaheuristics*. Lulu, segunda edição, 2013. Disponível em:

<<http://cs.gmu.edu/~sean/book/metaheuristics/>>. Acesso em: 20 de abr. de 2016.

RASMUSSEN, Rasmus V.; TRICK, Michael A. The Timetable Constrained Distance Minimization Problem. **Carnegie Mellon University**. Pittsburgh, p. 1-16. jun. 2006.

Disponível em:

<<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1511&context=tepper>>. Acesso em: 18 mar. 2017.

RIBEIRO, Celso C.; URRUTIA, Sebastián. Scheduling the Brazilian Soccer Tournament: Solution Approach and Practice. **Informis**. Maryland, USA, p. 260-272. maio 2012.

RIBEIRO, Celso C.. Sports scheduling: Problems and applications. **International Transactions In Operational Research**. [s. L.], p. 201-226. maio 2011.