

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIENCIAS EXATAS E APLICADAS

LUIZ GUSTAVO MOURA FERREIRA

**FORMULAÇÕES PARA PROBLEMAS DE AGENDAMENTO DE COMPETIÇÕES  
ESPORTIVAS**

João Monlevade

2016

LUIZ GUSTAVO MOURA FERREIRA

**FORMULAÇÕES PARA PROBLEMAS DE AGENDAMENTO DE COMPETIÇÕES  
ESPORTIVAS**

Monografia apresentada ao curso 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: George Henrique Godim da Fonseca

Coorientador: Paganini Barcellos de Oliveira

João Monlevade

2016



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, Aliz Gustavo Moura Feresira,  
declaro que o texto do trabalho de conclusão de curso intitulado  
“ FORMULAÇÕES PARA PROBLEMAS DE AGENDAMENTO DE  
COMPETIÇÕES ESPORTIVAS ” 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, 21 de Março de 2017

Aliz Gustavo Moura Feresira  
Assinatura do aluno



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 IV - Ata de Defesa

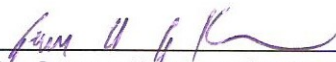
#### ATA DE DEFESA

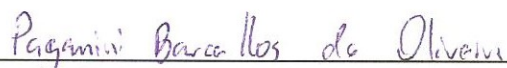
Aos dez dias do mês de Março de 2017, às dezenove horas e trinta minutos, na sala C203 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo aluno **LUIZ GUSTAVO MOURA FERREIRA**, sendo a Comissão Examinadora constituída pelos professores: Prof. Me. George Henrique Godim da Fonseca, Prof. Me. Paganini Barcelos de Oliveira, Prof. Dr. Rafael Frederico Alexandre e Prof. Me. Samuel Souza Brito.

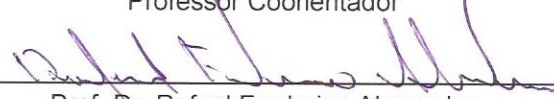
O candidato apresentou a monografia intitulada: "*Formulações para Agendamento de Competições Esportivas*". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 8,0 (oitos), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final.


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

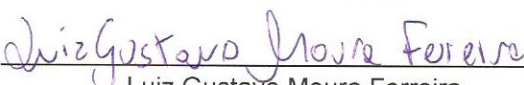
João Monlevade, 10 de Março de 2017.

  
\_\_\_\_\_  
Prof. Me. George Henrique Godim da Fonseca  
Professor Orientador/Presidente

  
\_\_\_\_\_  
Prof. Me. Paganini Barcelos de Oliveira  
Professor Coorientador

  
\_\_\_\_\_  
Prof. Dr. Rafael Frederico Alexandre  
Professor Convidado

  
\_\_\_\_\_  
Prof. Me. Samuel Souza Brito  
Professor Convidado

  
\_\_\_\_\_  
Luiz Gustavo Moura Ferreira  
Graduando

## **AGRADECIMENTOS**

Gostaria de agradecer a todos que participaram da minha caminhada durante longos anos de jornada até a formação acadêmica, agradeço ao meu amigo e orientador George Godim da Fonseca pela oportunidade de trabalharmos juntos na construção deste trabalho e pela sua orientação, estendo também meus agradecimentos ao meu coorientador e amigo Paganini Barcellos pela confiança e auxílio ao término do meu trabalho.

Agradeço a minha família, namorada e amigos que sempre me apoiaram e me trouxeram bastante confiança e alegria para que pudesse caminhar e chegar até meu objetivo.

\*\*\* As dificuldades não tornam apenas o percurso mais longo, mas a vitória mais empolgante, pois é persistindo que alcançamos o êxito \*\*\*

## RESUMO

Encontrar um calendário para competições esportivas é uma tarefa de grande importância econômica, pois as diversas viagens realizadas pelas equipes geram altos gastos de transporte e desgaste dos atletas. O *Traveling Tournament Problem (TTP)* é um problema de otimização combinatória da classe dos NP-difíceis que visa encontrar o melhor calendário para competições esportivas reduzindo a distância total viajada pelas equipes. Neste trabalho propõe-se uma formulação para minimizar os custos levando-se em conta as variações do TTP para problemas *Double Round Robin*, *Double Round Robin Unrestricted* e *Double Round Robin Mirrored*. O modelo é baseado em conjuntos independentes de grafos de conflitos. Para avaliar as formulações foram utilizados dados de problemas reais bem conhecidos da literatura obtidos no *Challenge Traveling Tournament Instances*. Através dos experimentos computacionais pôde-se observar que métodos exatos são capazes de solucionar apenas instâncias do problema com poucos times. Para quatro times a solução ótima é encontrada em poucos segundos, para seis e oito times soluções sub-ótimas podem ser encontradas em uma hora, e para dez ou mais times solução alguma é encontrada em uma hora. Assim, concluiu-se que métodos exatos ainda não são eficientes o suficiente para solucionar a maioria das instâncias desse problema.

**Palavras-chave:** *Traveling Tournament Problem*, *Double Round Robin*, Otimização Combinatória, Programação Inteira.

## ABSTRACT

To find a good schedule for sport competitions is a relevant task because of the high economical costs that incurs from the trips done by the participating teams and due to the fatigue of the athletes related to travelling. The Travelling Tournament Problem (TTP) is a NP-hard combinatorial optimization problem that aims at finding the best schedule for sport competitions reducing the total distance travelled by each team. In this work, Integer Programming formulations are proposed to solve some variations of this problem, such as Double Round Robin, Double Round Robin Unrestricted and Double Round Robin Mirrored. The formulation is based on finding independent sets in conflict graphs. Well known data from real world problems was used to evaluate the formulations. Computational experiments showed that the formulation is able to handle instances having a small number of teams. For instances having four teams the optimal solution is found within a few seconds, for instances having six and eight teams, sub-optimal solutions could be found within one hour, and, for instances having ten or more teams, no solution could be found within one hour of processing time. Therefore, it can be concluded that exact methods are still not effective enough to solve most of the real world instances of this problem.

**Key words:** *Traveling Tournament Problem, Double Round Robin, Combinatorial Optimization, Integer Programming.*



## LISTA DE FIGURAS

Figura 2.1 – Arvore Branch para o exemplo.....	18
Figura 2.2 – Efeito de redução Bound.....	19
Figura 3.1 - Grafo de Conflitos com 4 times.....	23

## LISTA DE TABELAS

Tabela 2.1 – Dois novos problemas gerados P1 e P2.....	18
Tabela 3.1 – Exemplo de agendamento de um campeonato espelhado com 4 times .....	24
Tabela 3.2 - Exemplo de agendamento de um campeonato não espelhado com 4 times .....	24
Tabela 4.1.1- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias NL .....	31
Tabela 4.1.2- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias CIRC .....	32
Tabela 4.1.3- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias SUPER .....	32
Tabela 4.1.4- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias GALAXY .....	32
Tabela 4.1.5- Resultado do Modelo Double Round Robin aplicado às instâncias NL .....	33
Tabela 4.1.6- Resultado do Modelo Double Round Robin aplicado às instâncias CIRC .....	33
Tabela 4.1.7- Resultado do Modelo Double Round Robin aplicado às instâncias SUPER .....	33
Tabela 4.1.8- Resultado do Modelo Double Round Robin aplicado às instâncias GALAXY .....	33
Tabela 4.1.9- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias NL .....	33
Tabela 4.1.10- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias CIRC .....	34
Tabela 4.1.11- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias SUPER .....	34
Tabela 4.1.12- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias GALAXY .....	34

## LISTA DE ABREVIATURAS

TTP	-	<i>Traveling Tournament Problem</i>
PLIM	-	Programação Linear Inteira Mista
LI	-	Limite Inferior
LS	-	Limite Superior
PPJ	-	Problema de Programação de Jogos
MTTP	-	<i>Mirrored Traveling Tournament Problem</i>
LB	-	<i>Lower Bound</i>
UB	-	<i>Upper Bound</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 PROBLEMA .....	14
1.2 OBJETIVOS .....	14
1.2.1 Objetivo geral .....	14
1.2.2 Objetivos específicos.....	15
1.4 ESTRUTURA DO TRABALHO.....	15
<b>2 CONCEITOS GERAIS E REVISÃO DA LITERATURA.....</b>	<b>16</b>
2.1 PROGRAMAÇÃO LINEAR E PROGRAMAÇÃO INTEIRA.....	16
2.1.1 Algoritmo Simplex .....	16
2.1.2 <i>Branch and Bound</i> .....	17
2.2 TRABALHOS RELACIONADOS .....	19
<b>3 PROBLEMA DO AGENDAMENTO DE COMPETIÇÕES ESPORTIVAS.....</b>	<b>22</b>
3.1 GRAFOS DE CONFLITOS.....	23
3.2 MODELO <i>DOUBLE ROUND ROBIN MIRRORED</i> .....	24
3.2.1 Parâmetros e Variáveis .....	25
3.2.2 Função objetivo e restrições.....	26
3.3 MODELO <i>DOUBLE ROUND ROBIN</i> .....	27
3.3.1 Os parâmetros e Variáveis .....	27
3.3.2 Função objetivo e restrições.....	29
3.3.3 Restrição de partidas consecutivas em casa e fora de casa.....	30
<b>4 EXPERIMENTOS COMPUTACIONAIS.....</b>	<b>31</b>
4.1 RESULTADOS OBTIDOS.....	31
4.1.1 Modelo <i>Double Round Robin Mirrored</i> .....	31
4.1.2 Modelo <i>Double Round Robin</i> .....	32
4.1.3 Modelo <i>Double Round Robin Unrestricted</i> .....	33
4.2 ANÁLISE DOS RESULTADOS .....	34

<b>5 CONCLUSÕES.....</b>	<b>36</b>
<b>REFERÊNCIAS.....</b>	<b>37</b>

# 1 Introdução

## 1.1 Problema

O problema de agendamento de competições esportivas consiste em encontrar uma tabela de confrontos entre os times participantes do campeonato de forma a minimizar os custos operacionais para as delegações, reduzir o desgaste dos atletas e promover equidade aos participantes, levando-se em conta vários fatores para a realização do ideal agendamento das partidas. O agendamento de partidas em competições esportivas possui grande importância teórica devido a se tratar de um problema NP-Difícil (WESTPHAL, 2011) e não existe algoritmo em tempo de execução polinomial para resolvê-lo.

Encontrar um calendário ideal para as competições reduzindo a distância percorrida entre as equipes é essencial, pois além de diminuir os gastos financeiros com viagens, também reduz o cansaço dos atletas, e uma vez que tais competições geram lucro para as equipes, as reduções desses custos podem refletir em campeonatos mais competitivos. Esse trabalho é focado em competições no formato *Double Round Robin*, em que cada equipe joga com duas vezes com cada uma das demais equipes, uma vez no primeiro turno e outra no segundo turno. Em cada turno cada equipe deve enfrentar todas as demais, uma por rodada, totalizando  $n - 1$  rodadas e  $\frac{n \cdot (n-1)}{2}$  jogos por turno. Easton, Nemhauser e Trick (2012), afirmam que o objetivo do TTP é encontrar um calendário do torneio *Double Round Robin* como objetivo de minimizar a distância total percorrida pelas equipes, satisfazendo também, as restrições específicas de cada torneio.

O presente trabalho apresenta formulações de Programação Linear Inteira Mista (PLIM) baseada em conjuntos independentes em um grafo de conflitos. No referido grafo, os vértices representam os jogos e as arestas indicam que esses jogos não podem ocorrer na mesma rodada. Algumas das instâncias do *Challenge Traveling Tournament* (TRICK, 2016) disponíveis na literatura, foram utilizadas para avaliar o desempenho das formulações implementadas.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O presente trabalho tem como objetivo solucionar o Problema de Agendamento de Competições Esportivas de forma exata utilizando Programação Linear Inteira Mista (PLIM). Adaptações à formulação serão feitas de modo a cobrir algumas variações do problema, como com turnos espelhados ou não, com ou sem restrição de sequência de

jogos em casa ou fora e com distâncias reais ou uniformes. A modelagem do problema visa avaliar a efetividade do modelo matemático frente a abordagens que constituem o estado da arte na literatura.

### **1.2.2 Objetivos específicos**

Como objetivos específicos do trabalho, pode-se citar:

- Avaliar métodos exatos de solução para o Problema de Agendamento de Competições Esportivas;
- Desenvolver modelos matemáticos para variações do problema; e
- Avaliar a dificuldade de resolução dessas variações.

### **1.4 Estrutura do trabalho**

O presente trabalho está organizado em 6 capítulos. O Capítulo 2 terá a fundamentação teórica do trabalho, como o uso de programação matemática e de otimização combinatória para a solução de problemas, além das metodologias abordadas para a resolução do problema. O Capítulo 3 apresentará o modelo do problema do agendamento de competições esportivas abordado bem como as formulações de PLIM para o problema e suas variações. O Capítulo 4 apresenta as instâncias utilizadas, os experimentos computacionais realizados e uma discussão dos resultados obtidos após a implantação da modelagem. No Capítulo 5 serão apresentadas as considerações finais e sugestões de trabalhos futuros.

## 2 Conceitos gerais e revisão da literatura

### 2.1 Programação Linear e Programação Inteira

De acordo com Hiller e Lieberman (2010), a programação linear é um grande e importante avanço tecnológico, pois desde sua criação em meados do século XX, chegando aos dias atuais vem economizando muito dinheiro para diversas empresas de países industrializados.

Para Martins (2013), o objetivo da programação linear é conseguir encontrar a melhor solução entre um conjunto com várias soluções para um problema. Para essa solução é dado o nome de solução ótima, e no que se refere ao termo linear, é devido às equações do modelo referido serem lineares. A melhor solução é definida de acordo com o problema que pode ser de maximização quando a solução refere-se ao maior valor encontrado da função objetivo, e um problema de minimização quando refere-se ao menor valor de função objetivo encontrado.

A estrutura de programação linear é utilizada nos problemas de otimização para descrever o problema e resolve-lo. A estrutura permite que se possa especificar, um conjunto de variáveis de decisão, conjunto de restrições às quais serão aplicadas sobre essas variáveis e um objetivo linear. A estrutura geral de problemas de programação linear é dada a seguir:

<b>Objetivo:</b>	minimizar $c'x$
<b>Restrições:</b>	$Ax \leq b$ (Restrições Lineares)
	$l \leq x \leq u$ (Restrições de limite)

Como vemos acima,  $x$  é um vetor que representa as variáveis de decisão,  $c$  a função objetivo linear, a equação  $Ax \leq b$  especifica as restrições lineares em  $x$ ,  $l$  e  $u$  são os limites inferior e superior em  $x$  (GUROBI, 2015).

#### 2.1.1 Algoritmo Simplex

Proposto por Dantzig (1947), o simplex foi o primeiro algoritmo para resolver problemas de programação linear. Apesar do tempo de existência deste algoritmo ele ainda continua sendo um dos métodos mais confiáveis e eficientes para resolver problemas lineares. Uma alternativa a este método é o algoritmo de barreira ou dos pontos interiores. Os métodos que utilizam pontos interiores têm beneficiado muito os recentes avanços na arquitetura de computadores, podendo assim incluir processadores *multi-core* e instruções SIMD sets, e são geralmente considerados como sendo mais rápido do que simplex para a



resolução de problemas de programação linear a partir do zero. Porém, mesmo com a existência de diversos modelos de Programação linear, e as diversas maneiras em que são utilizados, na prática nenhum algoritmo é melhor que o outro. Ambos são importantes na programação linear computacional (GUROBI, 2015).

O algoritmo simplex utiliza-se de ferramentas baseadas na álgebra linear para determinar através de métodos iterativos a melhor solução de Problema de Programação Linear. Sua geração inicial é simples e conseqüentemente eficiente e, assim, o algoritmo parte de uma solução viável dentro do sistema de equações que formam as restrições do problema linear. A partir dessa solução, o algoritmo busca por novas soluções viáveis de valor igual ou melhor que a atual solução. Basicamente o algoritmo possui dois critérios de escolha, um que permite encontrar sempre novas e melhores soluções para o problema, e outro para verificar se a solução escolhida é ou não a ótima (GOLDBARG e LUNA, 2005).

### 2.1.2 *Branch and Bound*

Os problemas onde ao menos uma variável de decisão é inteira são classificados como Programação Inteira. A técnica mais aplicada para resolver esse tipo de problema é o método *Branch-and-Bound*. Essa técnica utiliza-se dos conceitos de limite inferior e superior, onde o limite inferior representa o menor custo que uma solução pode ter e o limite superior é a melhor solução encontrada para o problema. O Gap é dado pela seguinte fórmula:  $1 - \frac{LI}{LS}$ .

O algoritmo de *Branch-and-Bound* se baseia na ideia de desenvolver uma enumeração inteligente de todos pontos que podem vir a serem solução ótima inteira de um dado problema. O termo *branch* se refere ao fato de que este método utiliza o espaço das soluções e efetua partições nele. O termo *bound* mostra que para se chegar na solução ótima utiliza-se de limites calculados ao longo da enumeração. (GOLDBARG e LUNA, 2005).

Para exemplificar, Goldberg e Luna (2005) utilizam o seguinte problema:

$$\text{Maximizar } z = 5x_1 + 8x_2$$

sujeito a:

$$x_1 + x_2 \leq 6$$

$$5x_1 + 9x_2 \leq 4$$

$$x_1, x_2 \in Z +$$

Para esse exemplo podemos verificar que a solução ótima do problema é encontrada em:  $x_1 = \frac{9}{4}$ ,  $x_2 = \frac{15}{4}$  levando a  $z = \frac{41}{4}$ . Partindo da ideia de separar a envoltória convexa em relação a  $x_2$ , podemos organizar a equação desta forma:

$$x_2 \geq \frac{15}{4} + 1 \geq 4 \quad x_2 \leq \frac{15}{4} \leq 3$$

A equação anterior produz duas restrições disjuntivas que, ao serem acrescentadas ao problema original geram dois novos problemas que não mais possuem a solução ótima contínua. Com isso, o problema original será reduzido a dois novos problemas, como apresentado na Tabela 2.1:

<p>(P<sub>1</sub>) Maximizar <math>z = cx</math>            sujeito a:  <math>A_x \leq b</math>  <math>x_i \in [x_i^0]</math>  <math>X_i \in Z</math></p>	<p>(P<sub>2</sub>) Maximizar <math>z = cx</math>            sujeito a:  <math>A_x \leq b</math>  <math>x_i \in [x_i^0] + 1</math>  <math>X_i \in Z</math></p>
---	---

Tabela 2.1 – Dois novos problemas P1 e P2

Com essa estratégia de separação cria-se novos e menores problemas que, serão de mais fácil solução. Neste exemplo, o problema (P) é separado em outros dois problemas (P1) e (P2) sendo  $x_2$  a variável escolhida para separação. Na árvore da Figura 2.1, cada nível representa uma separação ou *branch* em relação a uma variável.

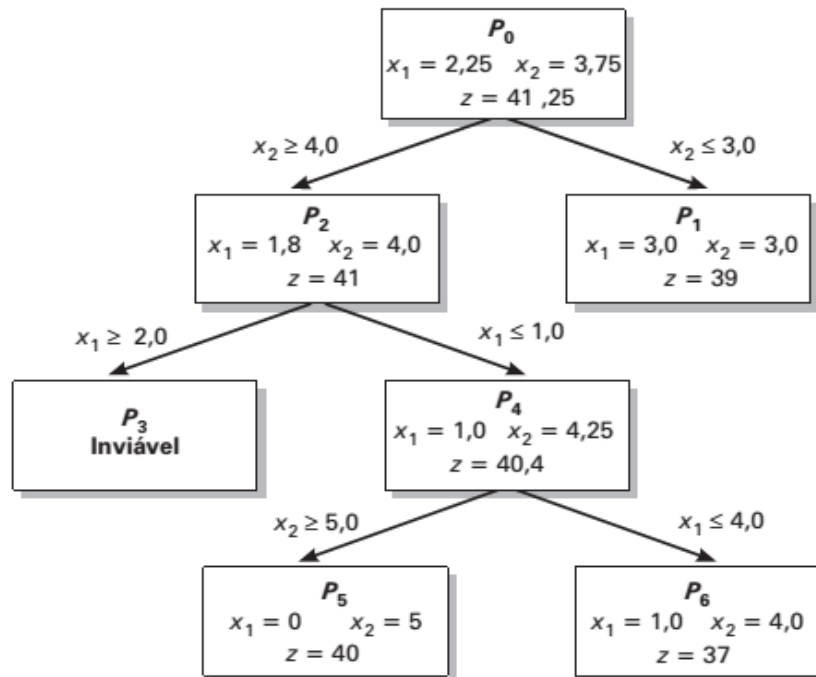


Figura 2.1 – Árvore de enumeração do problema exemplo. (GOLDBARG e LUNA, 2005, p. 182).

Para entender o efeito do *bound*, suponha sequência de *branches* da Figura 2.1, agora representada pela Figura 2.2, onde algumas soluções não precisarão serem exploradas, assim deixando de solucionar os problemas marcados.

Pela Figura 2.2 é possível notar que  $P_4$ , um problema com solução contínua, possui  $Z = 40,4$  e  $P_5$ , um problema com solução inteira, possui  $Z = 40$ , o problema  $P_6$  não precisa mais ser solucionado, pois entre 40,4 e 40 não existe outra solução inteira melhor que 40 ( $40 \leq z_0^* \leq 40,4$ ). O problema  $P_2$ , com  $z = 41$  pode dar origem, contudo, ainda a um problema com uma solução inteira de valor 41 ( $40 \leq z_0^* \leq 41$ ), o que obriga ao desenvolvimento de ( $P_3$ ). A redução pelo (*bound*) de apenas um vértice da árvore do exemplo parece pequena, mas temos que levar em conta que esse problema também é pequeno. Na realidade, a simplificação do limite inferior (ou superior no problema de minimização) tem poder significativo, sendo muito útil no processo de solução do problema.

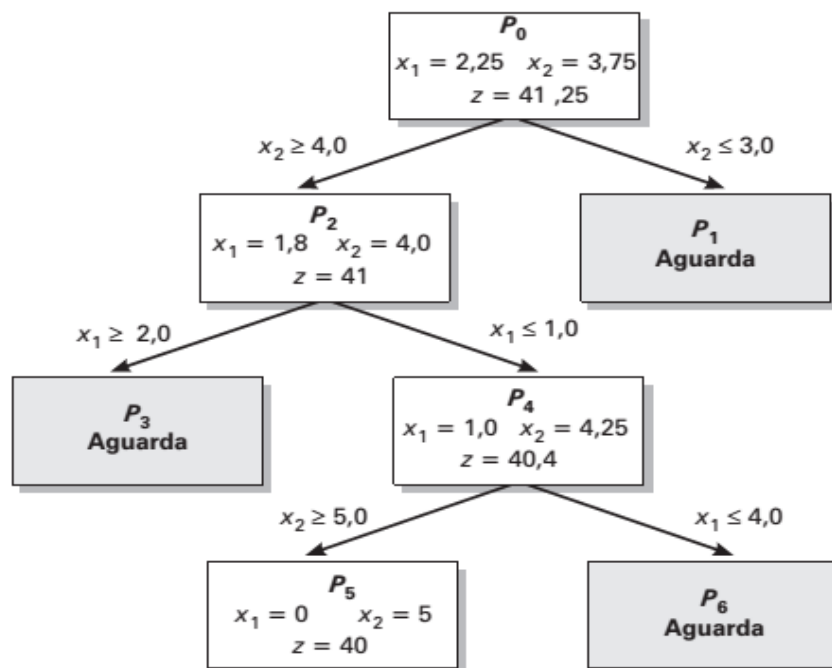


Figura 2.2 – Efeito de redução do *Bound* (GOLDBARG e LUNA , 2005, p. 183).

## 2.2 Trabalhos relacionados

O problema do agendamento de partidas, por se tratar de um problema de difícil resolução, vem sendo bastante pesquisado. Segundo Silva e Mine (2005), o Problema de Programação de Jogos de Competições Esportivas (PPJ), conhecido na literatura como *sports timetabling* ou *Traveling Tournament Problem* (TTP), é um problema pertencente à

classe de problemas NP-difíceis. Dessa forma, a obtenção da melhor solução existente por meio de métodos de programação matemática, ditos exatos, está limitada a problemas de pequenas dimensões.

O trabalho de Silva e Mine (2005) também aborda esse mesmo problema para o campeonato de futebol brasileiro com 24 equipes jogando turno e retorno que se enquadra no problema (MTTP), porém para tal trabalho o autor utilizou de abordagem heurística visto que a resolução por programação inteira não era viável para esse número de equipes.

Gonçalves (2014), aborda o problema de agendamento nas ligas americanas (MLB), e busca aplicar diferentes metaheurísticas como o ILS e Método do polígono para encontrar a solução objetivo do problema e comparar com outras soluções já encontradas no intuito de melhorar a precisão.

Urrutia, Ribeiro e Melo (2007), propõem em seu trabalho um método para melhorar as soluções para o *Traveling Tournament Problem*. O método consiste em aperfeiçoar o limite inferior independente, considerando as soluções ideais, se as distâncias entre todos os pares de cidades fossem constantes. Através desse método foi possível propor um novo *lower bound* para o TTP, o que possibilitou melhorar os resultados para diversas instâncias da competição internacional, e o benchmark do problema.

Easton, Nemhauser, e Trick (2003) abordam o TTP através de uma aplicação paralelizada do método *branch-and-price*, no qual o problema principal é resolvido por um método de geração de colunas e os subproblemas são resolvidos por programação de restrições. Não é considerada neste método a restrição de repetidores, onde são determinados o número de partidas em casa ou fora de casa. Os autores conseguem encontrar e provar a otimalidade para a a instancia NL8 em 4 dias, utilizando um computador de 20 processadores.

Em Cheung (2009) é utilizada a abordagem de decomposição de *Benders* para obter limites inferiores para o MTTP, propondo, adicionalmente um modelo de programação linear inteira mista. Melhores limites foram encontrados para as instâncias NL10 a NL16 e NFL16 a NFL24, exigindo de 3,5 a 22,5 dias de execução.

No trabalho de Carvalho e Lorena, (2012) é apresentada uma abordagem de programação matemática para o *Mirrored Traveling Tournament Problem* (MTTP) em um campeonato de times onde há o espelhamento, ou seja, os times jogam turno e retorno. Nesse tipo de campeonato o retorno é tomado em relação ao turno, porém com o mando de campo invertido, e o objetivo é diminuir a distância percorrida entre as equipes nesta competição. O trabalho sugere o uso de um grafo no qual as arestas adjacentes representam partidas que não podem acontecer na mesma rodada, e, a partir desse conjunto

de arestas, é calculada a função objetivo para o custo total percorrido. Para tal problema o autor utiliza-se de uma modelagem matemática para encontrar a melhor solução.

O presente trabalho utiliza técnicas baseadas na mesma idéia, onde as partidas são definidas a partir de um conjunto de arestas que definem as rodadas, porém serão propostos também, modelos para campeonatos *Double Round Robin* onde não há espelhamento das partidas, e, além disso, todas as instâncias utilizadas para os experimentos serão as mesmas adotadas no *Challenge Traveling Tournament* (TRICK, 2016).

### 3 Problema do Agendamento de Competições Esportivas

Na literatura, o problema do agendamento de competições esportivas é conhecido como *Travelling Tournament Problem* (TTP), e tem como objetivo construir uma tabela dos confrontos entre as equipes de uma competição.

Para a geração das competições esportivas deve-se levar em conta as diversas peculiaridades que ocorrem em cada campeonato e também a questão das distâncias percorridas entre as equipes, uma vez que são separadas geograficamente por grandes distâncias e necessitam se deslocar para a partida como visitante. Montar uma agenda de jogos é uma tarefa desafiadora, porém de grande valia.

O problema do agendamento de competições esportivas é de grande importância para instituições esportivas ao redor do mundo. Esse agendamento consiste em alocar todas as partidas de um campeonato atendendo a um conjunto de restrições dadas a priori com o menor percurso possível para as equipes. O problema de agendamento consiste em:

- Uma partida é realizada entre duas equipes, sendo uma das equipes jogando em casa e a outra fora.
- Uma rodada consiste em um conjunto de partidas, onde não se podem coincidir partidas com mesmo time.

Em relação às restrições, temos:

- Não poderá haver mais do que três jogos em casa ou três jogos fora consecutivos para qualquer equipe;
- *Double Round Robin*, ou seja, rodadas duplas, onde um determinado time A jogará com um time B em sua casa, posteriormente jogarão novamente, só que com os mandos de campo invertidos;
- Para N equipes serão necessárias  $2N-2$  Rodadas;
- Para cada participante, a diferença do número de jogos em seu domínio e fora dele é zero, ou seja, jogará o mesmo número de partidas em casa e fora de casa;
- Não poderá haver jogos consecutivos entre as equipes, ou seja, se duas equipes X,Y se enfrentam na rodada R, essas equipes não podem se enfrentar novamente na rodada R+1;
- Cada equipe pode jogar apenas uma vez por rodada;
- Todas as equipes devem jogar o mesmo número de partidas;

### 3.1 Grafos de Conflitos

A ideia do grafo de conflitos é que os vértices representam partidas, e essas partidas serão conflitantes se houver uma aresta que liguem elas, ou seja, os vértices são partidas, que são conflitantes por possuírem times em comum, e com isso não podem ser programadas para a mesma rodada.

Supondo-se quatro times - CAM, FLA, PAL e INT, de diferentes cidades brasileiras, que são respectivamente, Belo Horizonte, Rio de Janeiro, São Paulo e Porto Alegre. Para este número de equipes podemos totalizar 12 pares ordenados (partidas) que acontecerão em 6 rodadas distintas. O grafo de conflitos resultante é apresentado na Figura 3.1, onde podemos observar através das arestas adjacentes quais partidas não podem ser programadas para a mesma rodada por envolver times em comum. De acordo com o grafo podemos observar que as partidas entre PAL x FLA e INT x CAM podem acontecer na mesma rodada, já que não há arestas conflitantes ligando-os. Os modelos apresentados nas seções 3.2 e 3.3 se baseiam na ideia de detecção de conjuntos independentes em grafos em conflito.

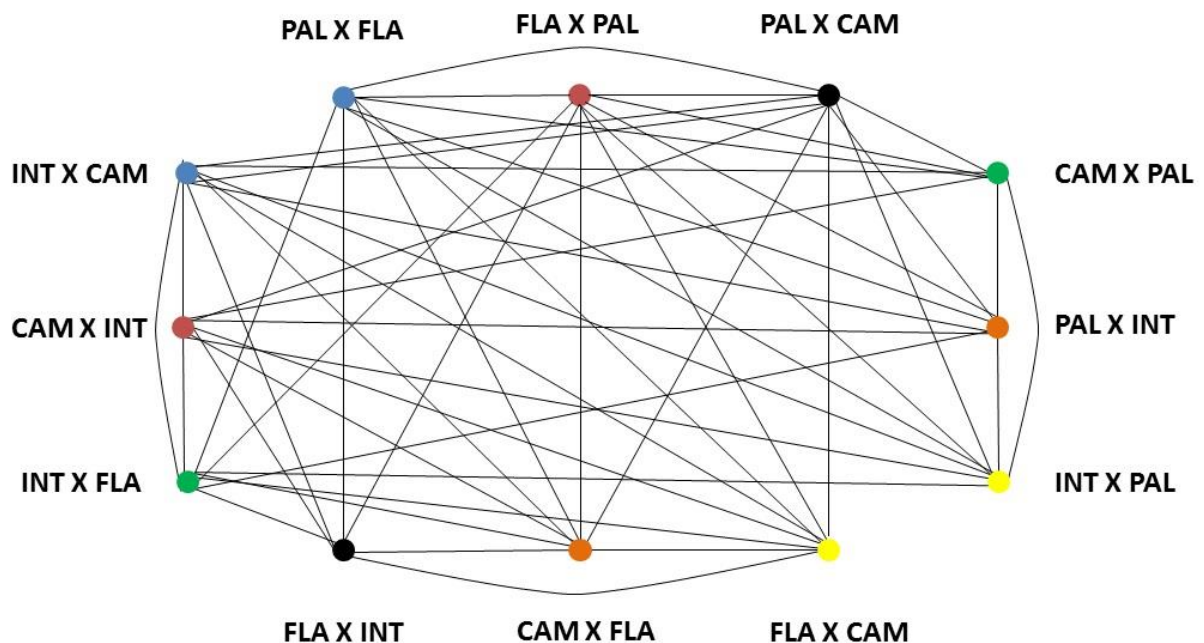


Figura 3.1 – Grafo de conflitos

Na Figura 3.1, temos que cada par de vértices da mesma cor representa partidas que podem ser disputadas em uma mesma rodada por não possuírem times em comum. Baseado nesse grafo de conflitos pode-se construir um agendamento para o torneio.

A Tabela 3.1 apresenta o agendamento de partidas levando-se em conta um campeonato espelhado.

Turno	Rodada	ID	Partida	Cidade
1	1	1	CAM X FLA	Belo Horizonte
		2	PAL X INT	São Paulo
	2	3	CAM X INT	Belo Horizonte
		4	PAL X FLA	São Paulo
	3	5	INT X FLA	Porto Alegre
		6	PAL X CAM	São Paulo
2	4	7	FLA X CAM	Rio de Janeiro
		8	INT X PAL	Porto Alegre
	5	9	INT X CAM	Porto Alegre
		10	FLA X PAL	Rio de Janeiro
	6	11	FLA X INT	Rio de Janeiro
		12	CAM X PAL	Belo Horizonte

Tabela 3.1- Exemplo de agendamento de partidas espelhado

Através do agendamento é possível verificar que o segundo turno do campeonato é idêntico ao primeiro, apenas invertendo o mando de campo das equipes. Como exemplo na Tabela 3.1, a partida entre CAM X FLA acontece na primeira rodada do primeiro turno e seu espelho acontece na primeira rodada do segundo turno. Um exemplo de torneio no qual não ocorre o espelhamento das partidas é apresentado na Tabela 3.2.

Turno	Rodada	ID	Partida	Cidade
1	1	1	CAM X FLA	Belo Horizonte
		2	PAL X INT	São Paulo
	2	3	CAM X INT	Belo Horizonte
		4	PAL X FLA	São Paulo
	3	5	INT X FLA	Porto Alegre
		6	PAL X CAM	São Paulo
2	4	7	INT X CAM	Porto Alegre
		8	FLA X PAL	Rio de Janeiro
	5	9	FLA X CAM	Rio de Janeiro
		10	INT X PAL	Porto Alegre
	6	11	FLA X INT	Rio de Janeiro
		12	CAM X PAL	Belo Horizonte

Tabela 3.2- Exemplo de agendamento de partidas sem espelhamento

Na tabela 3.2 é possível notar que as partidas que acontecem na primeira rodada do primeiro turno - CAM X FLA e PAL X INT - não acontecem na primeira rodada do segundo turno, o que descaracteriza um modelo espelhado.

### 3.2 Modelo *Double Round Robin Mirrored*

Em Carvalho e Lorena (2012), é proposta uma modelagem matemática que visa atender as necessidades do agendamento de competições esportivas em um campeonato



com espelhamento. Para este campeonato, o que se pretende é gerar um agendamento com melhor custo para o primeiro turno e o segundo turno é gerado pelo espelhamento do primeiro. O modelo proposto neste trabalho utiliza-se da técnica de grafos de conflitos, e sua função objetivo é minimizar a distância total percorrida entre as equipes. Tal modelo será descrito nessa seção.

### 3.2.1 Parâmetros e Variáveis

O modelo formulado para o trabalho segue os seguintes dados de entrada:

- $n$ : Número de times
- $p$ : Número de rodadas por turno (i.e.,  $n-1$ )
- $m$ : Número de possíveis pares ordenados de times que disputarão as partidas (i.e.,  $n \times (n-1)$ ). A partida e seu espelho são consecutivas.
- $E$ : Conjunto de arestas
- $g_i$ : Distância entre as cidades dos times participantes da partida  $i$
- $d_{i,j}$ : Distância entre partidas  $i$  e  $j$ , onde  $(i, j \in E)$
- $h_{i,j}$ : Distância entre partidas consecutivas  $i$  e  $j$ , onde  $(i, j \in E)$ , sendo que  $i$  é a última partida do primeiro turno e  $h$  é a primeira partida do segundo turno.

As variáveis do modelo são:

- $x_{i,k}$ : assume o valor 1 se a partida  $i$  ocorre na rodada  $k$  e assume 0 caso contrário ( $i = 1, \dots, m$  e  $k = 1, \dots, p$ ).
- $y_{i,k}$ : assume o valor 1 se a partida  $k$  ocorre na próxima rodada da partida  $i$  e assume 0 caso contrário ( $i = 1, \dots, m$  e  $k = 1, \dots, m$ ).
- $w_{i,k}$ : assume o valor 1 se a partida  $i$  ocorre na última rodada do primeiro turno e  $k$  na primeira rodada do segundo, assume 0 caso contrário.
- $s_i$ : assume o valor 1 se a partida  $i$  ocorre na primeira rodada do primeiro turno, assume 0 caso contrário .
- $e_i$ : assume o valor 1 se a partida  $i$  ocorre na última rodada do segundo turno, assume 0 caso contrário .

No modelo todas as variáveis são binárias. A primeira distância vai ser calculada através de  $g_i$ , que se refere à distância entre as cidades dos times participantes de determinada partida  $i$ , a fim de verificar os jogos da primeira rodada do primeiro turno e os jogos da última rodada do segundo turno, assim pode-se calcular o custo dos times que iniciam fora de casa e ao terminarem o turno retornam para casa.

A distância  $d_{i,j}$  refere-se à distância percorrida pelos times envolvidos nas partidas  $i$  e  $j$ . Por exemplo, suponha que se deseja calcular a distância entre as partidas 2 (PAL x INT) e 3 (CAM x INT) da Tabela 3.1. O clube INT teria de viajar de São Paulo a Belo Horizonte para atender a partida 3. Os times PAL e CAM não têm viagens envolvidas nas partidas 2 e 3. Se não houver times em comum entre as partidas, essa distância é zero.

O parâmetro  $h_{i,j}$  refere-se à distância entre partidas  $i$  e  $j$ , onde  $i$  é o último jogo do primeiro turno e  $j$  é o primeiro jogo do segundo turno. É possível tomar como exemplo as informações da Tabela 3.1, o qual a distância entre São Paulo e Rio de Janeiro, seriam representadas pelo parâmetro  $h_{6,7}$ .

### 3.2.2 Função objetivo e restrições

$$\text{Minimizar} \quad \sum_{(i,k) \in E} d_{i,k} y_{i,k} + \sum_{i=1}^m g_i s_i + \sum_{i=1}^m g_i e_i + \sum_{(i,k) \in E} h_{i,k} w_{i,k} \quad (1)$$

Sujeito a:

$$\sum_{j=1}^p (x_{i,j} + x_{i+1,j}) = 1 \quad i = \{1,3,5,\dots,m-1\} \quad (2)$$

$$x_{i,j} + x_{k,j} \leq 1 \quad \begin{array}{l} (i,k) \in E \\ j = 1,\dots,p \end{array} \quad (3)$$

$$x_{i,r} + x_{k,r+1} - y_{i,k} \leq 1 \quad \begin{array}{l} (i,k) \in E \\ r = 1,\dots,p \end{array} \quad (4)$$

$$x_{i,1} + x_{i+1,1} - s_i = 0 \quad i = \{1,3,5,\dots,m-1\} \quad (5)$$

$$x_{i,p} + x_{i+1,p} - e_i = 0 \quad i = \{1,3,5,\dots,m-1\} \quad (6)$$

$$x_{i,1} + x_{k,p} - w_{k,j} \leq 1 \quad (i,k) \in E \quad (7)$$

$$i = \{1,3, \dots, m-1\}$$

$$j = i+1$$

$$x_{a,j} + x_{b,j+1} + x_{c,j+2} + x_{f,j+3} \leq 3 \quad j = 1, \dots, p-3 \quad (8)$$

$$a = \{1..m\}, b = \{1..m\},$$

$$c = \{1..m\}, f = \{1..m\}.$$

$$x_{i,j} \in \{0,1\} \quad i = \{1, \dots, m\} \quad j = \{1, \dots, p\} \quad (9)$$

$$y_{i,k} \in \{0,1\} \quad (i,k) \in E \quad (10)$$

$$w_{i,k} \in \{0,1\} \quad (i,k) \in E \quad (11)$$

$$s_i \in \{0,1\} \quad i = \{1, \dots, m\} \quad (12)$$

$$e_i \in \{0,1\} \quad i = \{1, \dots, m\} \quad (13)$$

Para o modelo espelhado, a equação (1) refere-se à função objetivo e é composta por quatro somatórios que são: a distância entre duas partidas com times em comum, o custo de um time que se desloca de casa na primeira rodada do torneio, o custo para os times para voltarem para casa na última rodada e o custo entre a última rodada do primeiro turno e a primeira rodada do segundo turno. A equação (2) prevê que, se uma partida for agendada, a partida que representa seu espelho não poderá estar agendada no mesmo turno. Na equação (3) a detecção do conjunto independente de grafos não permite partidas conflitantes serem agendadas para a mesma rodada. As equações de (4) a (7) não permitem que a partida ou seu espelho ocorram: entre rodadas no mesmo turno (4), no início do primeiro turno (5), após a última rodada (6) e entre os dois turnos (7). A equação (8) limita a sequência de jogos em casa e fora. As equações (9) a (13) definem que as variáveis são binárias.

### 3.3 Modelo *Double Round Robin*

No trabalho é proposto um novo modelo de programação inteira para o agendamento de campeonatos no modelo *Double Round Robin*, modelo no qual não é requerido que o segundo turno seja espelhado em relação ao primeiro. Para tal o modelo baseia-se na mesma ideia de detecção de conjuntos independentes em grafos de conflitos.

#### 3.3.1 Os parâmetros e Variáveis

Os parâmetros de entrada para a formulação são:

- $n$ : Número de times
- $\hat{p}$ : Número de rodadas (i.e.,  $2n-2$ )
- $\hat{m}$ : Número de partidas (i.e.,  $n \times (n-1)$ ).

- $E$ : Conjunto de arestas
- $g_i$ : Distância entre as cidades dos times participantes da partida  $i$
- $d_{i,j}$ : Distância entre partidas  $i$  e  $j$ , onde  $(i,j \in E)$
- $H_i$ : Conjunto de partidas em que time  $T$  joga em casa
- $A_i$ : Conjunto de partidas em que time  $T$  joga fora de casa
- $f_m$ : Indica, para cada partida  $m$  a outra partida entre as mesmas equipes com o mando de campo invertido.

As variáveis de decisão do modelo são:

- $x_{i,k}$ : assume o valor 1 se a partida  $i$  ocorre na rodada  $k$  e assume 0 caso contrário ( $i = 1, \dots, \hat{m}$  e  $k = 1, \dots, \hat{p}$ ).
- $y_{i,k}$ : assume o valor 1 se a partida  $k$  ocorre na próxima rodada da partida  $i$  e assume 0 caso contrário ( $i = 1, \dots, \hat{m}$  e  $k = 1, \dots, \hat{m}$ ).
- $S_i$ : assume o valor 1 se a partida  $i$  ocorre na primeira rodada e assume 0 caso contrário .
- $e_i$ : assume o valor 1 se a partida  $i$  ocorre na última rodada e assume 0 caso contrário .

No modelo todas as variáveis assumem a forma binária, e pode-se observar que para calcular a distância percorrida para cada equipe durante o torneio leva-se em conta onde a equipe está a cada rodada e para onde irá à próxima rodada.

A primeira distância a ser calculada refere-se a  $g_i$ , onde é calculada a distância entre as cidades dos times participantes daquela determinada partida  $i$ , sua utilidade é verificar se a partida pertence à primeira rodada de um turno ou a última rodada deste turno. Assim ao iniciar o segundo turno sabe-se qual time terá de se deslocar para a partida, e também é possível calcular o custo dos times que iniciam o turno fora de casa e quando terminam, voltam para casa.

No modelo *Double Round Robin* são criados dois agendamentos que correspondem a dois turnos do campeonato, cada qual com sua configuração de partidas, que diferente do modelo espelhado, não precisa ser necessariamente igual ao primeiro turno.

A distância definida por  $d_{i,j}$  refere-se à soma da distância a ser percorrida pelos times envolvidos nas partidas  $i$  e  $j$  caso essas partidas ocorram em rodadas consecutivas. Na Tabela 3.2, um exemplo é  $d_{2,4}$  que corresponde à distância que o time PAL terá de percorrer entre as partidas 2 e 4 que, neste caso, é igual a zero devido a estar na mesma

cidade. Tomando como base novamente a Tabela 3.2, nota-se que da primeira para a segunda rodada do primeiro turno, o time INT irá percorrer a distância  $d_{2,3}$  relativa ao deslocamento da cidade de São Paulo para Belo Horizonte.

Os conjuntos  $H_t$  e  $A_t$  se referem no modelo às restrições de partidas consecutivas onde o time  $t$  não pode disputar mais de três partidas consecutivas em casa ou fora de casa. Já o parâmetro  $\hat{r}_m$  é utilizado para que não haja confrontos repetidos entre as equipes na próxima rodada, ou seja, supondo que  $\hat{r}_i = j$ , caso a partida  $i$  ocorra na rodada  $r$ , à partida  $j$  não pode ocorrer na rodada  $r + 1$ .

### 3.3.2 Função objetivo e restrições

O modelo *Double Round Robin*, que tem como função objetivo minimizar a distância percorrida entre as equipes no agendamento das partidas é apresentado em sequência.

$$\text{Minimizar} \quad \sum_{(i,k) \in E} d_{i,k} y_{i,k} + \sum_{i=1}^m g_i s_i + \sum_{i \in E} g_i e_i \quad (14)$$

$$\text{Sujeito a:} \quad \sum_{r \in R} x_{m,r} = 1 \quad m \in E \quad r = \{1, \dots, \hat{p}\} \quad (15)$$

$$x_{i,r} + x_{k,r} \leq 1 \quad (i,k) \in E, r \in R, r = \{1, \dots, \hat{p}\} \quad (16)$$

$$x_{i,r} + x_{k,r+1} - y_{i,k} \leq 1 \quad (i,k) \in E, r \in R, r \neq \hat{p} \quad (17)$$

$$x_{i,1} = s_i \quad i \in M, i = \{1, \dots, \hat{m}-1\} \quad (18)$$

$$x_{i,\hat{p}} = e_i \quad i \in M, i = \{1, \dots, \hat{m}-1\} \quad (19)$$

$$x_{i,r} + x_{\hat{r}_i, r+1} \leq 1 \quad i \in M \quad r \in R, r \neq \hat{p} \quad (20)$$

$$x_{i,r} \in \{0,1\} \quad i \in M, i = \{1, \dots, \hat{m}\}, r \in R, r = \{1, \dots, \hat{p}\} \quad (21)$$

$$y_{i,k} \in \{0,1\} \quad (i,k) \in E \quad (22)$$

$$s_i \in \{0,1\} \quad i \in M, i = \{1, \dots, \hat{m}\} \quad (23)$$

$$e_i \in \{0,1\} \quad i \in M, i = \{1, \dots, \hat{m}\} \quad (24)$$

No modelo descrito pelas equações de (14) à (24), a equação (14) refere-se à função objetivo e é composta por três somatórios que consideram respectivamente, a distância entre duas partidas com times em comum, o custo de um time que se desloca de

casa na primeira rodada do torneio e por fim no último somatório temos o custo para os times para voltarem  $i$  para casa na última rodada.

A equação (15) garante que cada partida será atribuída a uma rodada, ou seja, cada qual pertence a um dos turnos. A equação (16) utiliza o conjunto independente de grafos para não permitir partidas conflitantes na mesma rodada. A equação (17) ativa a variável  $y_{i,k}$ , indicando que as partidas  $i$  e  $k$  ocorrem em rodadas consecutivas, a equação (18) ativa a variável  $s_i$  caso a partida  $i$  ocorra na primeira rodada e a equação (19) ativa  $e_i$  caso a partida  $i$  ocorra na última rodada. A equação (20) restringe que a partida que ocorreu na rodada  $r$  não aconteça na rodada  $r+1$ . As equações de (21) à (24) definem as variáveis binárias.

### 3.3.3 Restrição de partidas consecutivas em casa e fora de casa

O modelo inscrito pelas equações de 14 a 24, é uma variação do modelo *Double Round Robin* onde não há restrições de partidas. O modelo o qual não há restrições de partidas será referido nesse trabalho como *Double Round Robin Unrestricted*. Para tornar o modelo restrito, onde há restrição de partidas consecutivas em casa e fora, serão adicionadas as seguintes restrições ao modelo:

$$x_{m_1,r} + x_{m_2,r+1} + x_{m_3,r+2} + x_{m_4,r+3} \leq 3 \quad (m_1, m_2, m_3, m_4) \in H_t, r \in R, t \in T \quad (25)$$

$$x_{m_1,r} + x_{m_2,r+1} + x_{m_3,r+2} + x_{m_4,r+3} \leq 3 \quad (m_1, m_2, m_3, m_4) \in A_t, r \in R, t \in T \quad (26)$$

Essas restrições serão aplicadas ao modelo e com isso só serão permitidos agendamentos nos quais o número de partidas consecutivas em casa ( $H_t$ ) ou fora de casa ( $A_t$ ) seja menor ou igual a três. Nas restrições (25) e (26)  $(m_1, m_2, m_3, m_4)$  representam partidas que violam a restrição caso agendadas em rodadas consecutivas.

## 4 Experimentos Computacionais

Para os experimentos do presente trabalho foram utilizadas as instâncias do *Challenge Traveling Tournament* e um computador equipado com Processador Intel Core i3-2370M com 2.4GHz e 4GB de memória RAM, com sistema Operacional *Windows 7 Ultimate*. Os modelos foram resolvidos utilizando o Gurobi 6.5.0.

### 4.1 Resultados Obtidos

Iniciou-se os experimentos com o modelo *Double Round Robin Mirrored*, *Double Round Robin* e em seguida o modelo *Unrestricted*, e para tal foram analisados os seguintes aspectos em relação aos resultados: Limite inferior, Limite superior, *Gap*, número de restrições (*Linhas*), de variáveis (*Colunas*) e quantidade de Não-Zeros (NZ). Assim pôde-se analisar os resultados obtidos a partir dos experimentos e o tamanho dos problemas. Nas tabelas a seguir, a coluna LS\* possui os valores das melhores soluções conhecidas (UB) até então para as respectivas instâncias.

Para a realização dos experimentos foram considerados os instâncias com até 10 times e os tempos de execuções para todas as instâncias de 1 hora, tempo menor em relação a diversos trabalhos observados na área dos métodos exatos, devido à indisponibilidade de tempo para experimentos mais longos.

#### 4.1.1 Modelo *Double Round Robin Mirrored*

As Tabelas de 4.1.1 à 4.1.4 apresentam os resultados obtidos a partir dos experimentos realizados com o modelo *Double Round Robin Mirrored*.

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
NL4	8.276	8.276	0,0%	120	3.228	6.430	8.276
NL6	6.320	29.274	84,17%	512	62.550	138.194	26.568
NL8	5.636	57.140	94,89%	1.350	461.272	1.050.922	41.928
NL10	-	-	-	2.630	2.074.050	4.786.252	58.190

Tabela 4.1.1- Resultado do Modelo *Double Round Robin Mirrored* aplicado às instâncias NL

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>CIRC4</b>	20	20	0,0%	120	3.228	6.430	20
<b>CIRC 6</b>	18	79	84,26%	512	62.550	138.194	72
<b>CIRC 8</b>	14	191	96,15%	1.350	461.272	1.050.922	140
<b>CIRC 10</b>	-	-	-	2.630	2.074.050	4.786.252	280

**Tabela 4.1.2- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias CIRC**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>SUPER4</b>	63.405	63.405	0,0%	120	3.228	6.430	-
<b>SUPER 6</b>	18.629	146.697	87,09%	512	62.550	138.194	-
<b>SUPER 8</b>	16.018	317.210	96,12%	1.350	461.272	1.050.922	-
<b>SUPER 10</b>	-	-	-	2.630	2.074.050	4.786.252	-

**Tabela 4.1.3-Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias SUPER**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>GALAXY4</b>	416	416	0.0%	120	3.228	6.430	-
<b>GALAXY6</b>	323	1.578	81.45%	512	62.550	138.194	-
<b>GALAXY8</b>	183	3.297	94.88%	1.350	461.272	1.050.922	-
<b>GALAXY10</b>	-	-	-	2.630	2.074.050	4.786.252	-

**Tabela 4.1.4- Resultado do Modelo Double Round Robin Mirrored aplicado às instâncias GALAXY**

#### 4.1.2 Modelo *Double Round Robin*

As Tabelas de 4.1.5 à 4.1.8 apresentam os resultados obtidos a partir dos experimentos realizados com o modelo *Double Round Robin*.

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>NL4</b>	8.276	8.276	0,0%	204	3.228	10.932	8.276
<b>NL6</b>	4.231	26.728	84,17%	870	62.550	234.930	22.969
<b>NL8</b>	2.686	52.640	94,89%	2.296	461.272	1.786.568	38.760
<b>NL10</b>	-	-	-	4.770	2.074.050	8.136.630	56.506

**Tabela 4.1.5- Resultado do Modelo Double Round Robin aplicado às instâncias NL**



Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>CIRC4</b>	20	20	0,0%	204	3.228	10.932	20
<b>CIRC 6</b>	12	74	83,78%	870	62.550	234.930	64
<b>CIRC 8</b>	8	184	95,65%	2.296	461.272	1.786.568	128
<b>CIRC 10</b>	-	-	-	4.770	2.074.050	8.136.630	384

**Tabela 4.1.6- Resultado do Modelo Double Round Robin aplicado às instâncias CIRC**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>SUPER4</b>	63.405	63.405	0,0%	204	3.228	10.932	63.405
<b>SUPER 6</b>	17.824	145.844	87,77%	870	62.550	234.930	130.365
<b>SUPER 8</b>	14.466	315.569	95,41%	2.296	461.272	1.786.568	182.409
<b>SUPER 10</b>	-	-	-	4.770	2.074.050	8.136.630	316.329

**Tabela 4.1.7- Resultado do Modelo Double Round Robin aplicado às instâncias SUPER**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>GALAXY4</b>	416	416	0,0%	204	3.228	10.932	416
<b>GALAXY6</b>	300	1.509	80,11%	870	62.550	234.930	1.365
<b>GALAXY8</b>	174	3.276	94,68%	2.296	461.272	1.786.568	2.373
<b>GALAXY10</b>	-	-	-	4.770	2.074.050	8.136.630	4.554

**Tabela 4.1.8- Resultado do Modelo Double Round Robin aplicado às instâncias GALAXY**

#### **4.1.3 Modelo *Double Round Robin Unrestricted***

As Tabelas de 4.1.1 à 4.1.4 apresentam os resultados obtidos a partir dos experimentos realizados com o modelo *Unrestricted*.

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>NL4</b>	8.276	8.276	0,0%	204	1.284	3.156	-
<b>NL6</b>	6.031	20.769	70,96%	870	10.050	24.930	-
<b>NL8</b>	3.327	41.011	91,88%	2.296	38.696	96.264	-
<b>NL10</b>	3.300	67.864	95,13%	4.770	105.750	263.430	-

**Tabela 4.1.9- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias NL**

	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>CIRC4</b>	20	20	0,0%	204	1.284	3.156	-
<b>CIRC 6</b>	15	56	73,21%	870	10.050	24.930	-
<b>CIRC 8</b>	10	110	90,90%	2.296	38.696	96.264	-
<b>CIRC 10</b>	11	296	96,28%	4.770	105.750	263.430	-

**Tabela 4.1.10- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias CIRC**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>SUPER4</b>	63.405	63.405	0,0%	204	1.284	3.156	-
<b>SUPER 6</b>	34.018	113.356	69,99%	870	10.050	24.930	-
<b>SUPER 8</b>	15.910	197.398	91,94%	2.296	38.696	96.264	-
<b>SUPER 10</b>	4.365	597.943	99,27%	4.770	105.750	263.430	-

**Tabela 4.1.11- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias SUPER**

Instância	LI	LS	Gap	Colunas	Linhas	NZ	LS*
<b>GALAXY4</b>	416	416	0,0%	204	1.284	3.156	-
<b>GALAXY6</b>	339	1.199	71,72%	870	10.050	24.930	-
<b>GALAXY8</b>	228	2.659	91,42%	2.296	38.696	96.264	-
<b>GALAXY10</b>	253	5.192	95,12%	4.770	105.750	263.430	-

**Tabela 4.1.12- Resultado do Modelo Double Round Robin Unrestricted aplicado às instâncias GALAXY**

## 4.2 Análise dos Resultados

Os modelos resolvem problemas de 4 times sem dificuldades em todas as variações, enquanto para instâncias um pouco maiores, como por exemplo instâncias com 6 e 8 times, excedeu-se o tempo de execução e não foi encontrada a solução ótima, todavia encontram soluções próximas às melhores conhecidas na literatura. Nota-se também a partir dos resultados que para instâncias com 10 times o modelo não foi capaz de produzir solução alguma, enquanto que modelos com mais de 10 times levam a estouro de memória e, por isso não foram reportados nas tabelas. Outro fato a se observar é que o número de variáveis, restrições e Não-Zeros cresce vertiginosamente com o aumento do número de times. As soluções encontradas apresentam um Gap alto, isso é devido ao LI ser muito baixo, porém as soluções encontradas são próximas às encontradas na literatura.

Esse grande crescimento do tempo mostra quão difícil é resolver o problema de forma exata, e isso é devido à natureza do problema, que é da classe NP-difícil, assim de crescimento de complexidade e dificuldade de resolução exponencialmente proporcional ao tamanho da entrada.

Observa-se também a dimensão e complexidade das instâncias através do seu número de Linhas, Colunas e (NZ), que denominam respectivamente o número de restrições, variáveis e não zeros do problema linear. Esse alto nível de complexidade de resolução pode ser observado tomando como exemplo, a instância NL6 apresentada na Tabela 4.1.5, o qual se tem que o número de restrições 62550 e o número de variáveis 870, superior se comparado com a instância NL4 onde o número de restrições é apenas 3228 e o número de variáveis é 204. Este mesmo efeito pode ser observado nas Tabelas de 4.1.1 à 4.1.12 para as variações *Double Round Robin Mirrored* e *Unrestricted*.

Comparando-se os resultados obtidos com a implementação do modelo *Unrestricted*, é possível perceber grandes diferenças no tamanho dos problemas, para a mesma instância o problema diminui consideravelmente. Como exemplo, comparando os resultados obtidos pelas instâncias CIRC6 nas Tabelas 4.1.6 e 4.1.10, observa-se que o tamanho do problema corresponde a 870 variáveis e 62550 restrições na tabela 4.1.6, e 870 variáveis e 10050 restrições na Tabela 4.1.10. O que ocorre nas demais Tabelas 4.1.7 e 4.1.8 se comparado às Tabelas 4.1.11 e 4.1.12.

Esse comportamento observado entre os modelos restrito e irrestrito ocorrem devido à dificuldade de atendimento das restrições (25) e (26) do modelo, que faz com que o modelo se torne restrito ou irrestrito. Para o atendimento dessa restrição o problema cresce muito e com isso aumenta consumo de memória. Como exemplo, para instâncias com 10 times o problema se tornou tão grande que não foi possível encontrar nenhum limite para o mesmo, o qual se confirma ao se comparar as instâncias de 10 times nas Tabelas 4.1.1 a 4.1.12. Foi possível observar também que apenas o modelo *Unrestricted* consegue encontrar limites de LS e LI dentro do tempo estabelecido, devido a uma redução no tamanho de restrições do problema.

As Tabelas 4.1.1 e 4.1.5 apresentam o impacto da diferença do modelo espelhado para o modelo não espelhado, onde o número de variáveis é menor para o problema espelhado. Sabendo que o modelo espelhado agenda um turno do campeonato e o outro é tomado em relação a este agendamento, diferente do modelo não espelhado onde é agendado os dois turnos e, com isso, se tem um maior conjunto de variáveis.

## 5 Conclusões

As variações estudadas neste trabalho enfocam os casos dos torneios com dois turnos: sem espelhamento, irrestrito e com espelhamento. Em suma, para o presente trabalho foram desenvolvidas duas modelagens para as variações dos modelos *Double Round Robin* e *Unrestricted* além implementação do Modelo *Mirrored* de Lorena e Carvalho (2012). Ambos se baseiam na obtenção de conjunto independente grafo em conflitos, onde os vértices representam partidas, que são adjacentes, e que devido ao fato de terem times em comum não podem ser programadas para a mesma rodada.

Os modelos propostos têm como objetivo minimizar a soma das distâncias percorridas pelas equipes durante o campeonato. Os modelos *Double Round Robin* e *Double Round Robin Mirrored* utilizam a restrição de jogos consecutivos em casa ou fora de casa, e o modelo *Unrestricted* possui as mesmas característica exceto a utilização da restrição de partidas consecutivas.

Nos experimentos computacionais buscou-se avaliar o tamanho do problema gerado para cada uma das instâncias, bem como o Limite Inferior e Limite Superior para avaliar a qualidade das soluções encontradas e também os *benchmarks* da literatura. Com os dados obtidos pôde-se verificar a dificuldade de se resolver os problemas, e também que o tempo de execução cresce exponencialmente de acordo com o número de times.

Para instâncias com 4 times os modelos foram efetivos e encontraram a solução ótima, para instâncias de 6 e 8 times foi possível encontrar soluções próximas as encontradas na literatura e para instâncias maiores que 8 times os métodos implementados não foram efetivos para solucionar os problemas, isso devido à sua natureza combinatória.

Para trabalhos futuros, sugere-se utilizar técnicas de metaheurísticas quando o número de times é maior que 6, o que é verdade para a maioria dos problemas reais, assim podendo obter resultados melhores. Uma relaxação lagrangeana, como proposto por Benoist, Laburthe, e Rottembourg (2001), pode ser vista também como uma boa alternativa para a tentativa de resolução do problema, tendo em vista que neste método é proposto um relaxamento que aproxima um problema difícil de otimização a um problema simples, criando-se subproblemas para cada equipe. Variações dessa técnica como o método Langrangiano Aumentado proposto por Martínez (2009) também é opção de implementação. O uso de *Matheuristics*, (Boschetti, Roffilli, Maniezzo e Rohler, 2009) também tem potencial para ser explorado, uma vez que, nessa técnica ocorre a interoperação entre metaheurísticas e programação matemática.

## Referências.

- BENCHMARK. Travel Tournament Problem: Description and benchmarks. 2016. Disponível em: <<http://mat.gsia.cmu.edu/TOURN/>>. Acesso em: 1 jul 2016
- BENOIST, T., LABURTHE, F., e ROTTEMBOURG, B. (2001). Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems, Wye College (pp. 15–26).
- BOSCHETTI M., ROFFILLI M., MANIEZZO V. e ROHLER A. Matheuristics: Optimization, Simulation and Control. Hybrid Metaheuristics. Udine: Springer Berlin Heidelberg, 2009
- CHEUNG, K. K. H. A Benders approach for computing lower bounds for the mirrored traveling tournament problem. *Discrete Optimization*, 6(2), 189–196, Maio, 2009.
- DANTZIG G. Origins of the simplex method. A history of scientific computing, New York, ACM, 2012.
- EASTON, K., NEMHAUSER, G., & TRICK, M. Solving the travelling tournament problem: A combined integer programming and constraint programming approach. *Lect. notes comput. sci.* (Vol. 2740, pp. 100–109). Berlin, Heidelberg: Springer, 2003.
- GOLDBARG, LUNA. Otimização Combinatória e Programação Linear. Rio de Janeiro, Elsevier Editora Ltda., 2005.
- GONÇALVES, G. Abordagens Heurísticas para o Problema de Agendamento de Jogos de Competições Esportivas – DECEA, Universidade Federal de Ouro Preto, João Monlevade, 2014.
- GUROBI, Linear Programming (LP) - A Primer on the Basics, 2016. Disponível em <<http://www.gurobi.com/resources/getting-started/lp-basics>>. Acesso em: 10 mai 2016.
- HILLIER, F. S.; LIEBERMAN, G. J. Introdução à pesquisa operacional. 9. ed. Porto Alegre, AMGH, 2012.
- KIM, B. M. Iterated Local Search for the Traveling Tournament Problem. Tese (Doutorado) - Technische Universität Wien, 2012.
- LORENA, L., CARVALHO, M. New models for the Mirrored Traveling Tournament Problem. *Computers & Industrial Engineering*, 63, 1089-1095, agosto, 2012.
- MARTINEZ, J. Otimização prática usando o langraniano aumentado, IMECC, Unicamp, São Paulo, 2009.

MARTINS, A. G. Simulação das operações de lavra da Mina de Brucutu utilizando um modelo de programação linear para alocar os equipamentos de carregamento. Dissertação (Mestrado em Engenharia Mineral) - Universidade Federal de Ouro Preto, Ouro Preto, 2013.

SILVA G.; MINE, M. S. M. Um método de geração de uma solução inicial baseado em backtracking para o problema de programação de jogos do campeonato brasileiro de futebol – DECOM, Universidade Federal de Ouro Preto, Ouro Preto, 2005

TRICK, Michael Trick's Operations Research Page. Disponível em <<http://mat.tepper.cmu.edu/TOURN/>>. Acesso em: 10 out 2016.

URRUTIA, S., RIBEIRO, C. C., & MELO, R. A. A new lower bound to the traveling tournament problem. In Proceedings of the IEEE Symp. on Comput. Intell. in Sched. Honolulu (pp. 15–18), 2007.

WESTPHAL, C. Complexity of the Traveling Tournament Problem – Departamento de Matemática, University of Kaiserslautern, Kaiserslautern, 2011.