

DEPARTAMENTO DE COMPUTAÇÃO
DECOM

Um Método de Geração de uma Solução Inicial
baseado em *Backtracking* para o Problema de
Programação de Jogos do
Campeonato Brasileiro de Futebol

Matheus de Souza Alves Silva
Marcio Tadayuki Mine
Marcone Jamilson Freitas Souza

Relatório Técnico DECOM 10/2005

Julho / 2005

U F O P
UNIVERSIDADE FEDERAL DE OURO PRETO

Resumo

Este trabalho apresenta uma metodologia para gerar uma solução inicial aleatória para o problema de programação de jogos da primeira divisão do Campeonato Brasileiro de Futebol, o qual é realizado em dois turnos completos e espelhados. O método consiste em gerar uma solução inicial por um procedimento de duas fases, cada qual baseada em *backtracking*. Na primeira fase são geradas apenas as duas primeiras e as duas últimas rodadas de cada turno, enquanto que a segunda fase gera todas as demais rodadas intermediárias. Com esse método, pode-se gerar uma solução inicial satisfazendo-se a duas restrições do problema abordado.

Índice

RESUMO.....	2
ÍNDICE.....	3
LISTA DE TABELAS.....	4
1 INTRODUÇÃO	5
2 O PROBLEMA DE PROGRAMAÇÃO DE JOGOS ABORDADO	6
3 METODOLOGIA.....	8
3.1 REPRESENTAÇÃO DE UMA SOLUÇÃO	8
3.2 FUNÇÃO DE AVALIAÇÃO	9
3.3 GERAÇÃO DE UMA SOLUÇÃO INICIAL.....	9
4 CONCLUSÃO.....	16
REFERÊNCIAS BIBLIOGRÁFICAS	17

Lista de Tabelas

Tabela 1: Exemplo de uma escala de jogos do PPJE	6
Tabela 2: Exemplo de uma solução na representação de Anagnostopoulos <i>et al.</i> (2003).....	8
Tabela 3: Matriz base inicial	10
Tabela 4: Matriz de domínios.....	10
Tabela 5: 1ª Iteração	11
Tabela 6: Matriz de domínios atualizada para a 1ª iteração	11
Tabela 7: 2ª Iteração – Alocação do oponente do time 2	11
Tabela 8: 3ª Iteração – Alocação do oponente do time 3	11
Tabela 9: 4ª Iteração – Alocação do oponente do time 4	12
Tabela 10: 5ª Iteração – Alocação do oponente do time 5	12
Tabela 11: 6ª Iteração – Alocação do oponente do time 6	12
Tabela 12: 7ª Iteração – <i>Backtracking</i> 1	12
Tabela 13: 8ª Iteração – <i>Backtracking</i> 2.....	13
Tabela 14: 11ª Iteração – Alocação do oponente do time 8	13
Tabela 15: 1ª Iteração – Alocação do oponente do time 1 para a terceira rodada.....	13
Tabela 16: Rodadas geradas pela primeira fase do método	14
Tabela 17: Configuração da escala ao final da primeira fase	14
Tabela 18: Configuração da escala ao final da segunda fase	14
Tabela 19: Solução inicial	15

1 Introdução

Este trabalho apresenta uma metodologia para gerar uma solução inicial aleatória para o problema de programação de jogos da primeira divisão do Campeonato Brasileiro de Futebol, o qual é realizado em dois turnos completos e espelhados. O método consiste em gerar uma solução inicial por um procedimento de duas fases, cada qual baseada em *backtracking*. Na primeira fase são geradas apenas as duas primeiras e as duas últimas rodadas de cada turno, enquanto que a segunda fase gera todas as demais rodadas intermediárias. Com esse método, pode-se gerar uma solução inicial satisfazendo-se a duas restrições do problema abordado.

Este trabalho está estruturado em quatro capítulos, incluindo esta introdução.

No capítulo 2 é descrito o problema de programação de jogos do Campeonato Brasileiro de Futebol realizado em dois turnos espelhados.

No capítulo 3 é apresentada a metodologia de geração de uma solução inicial aleatória e o capítulo 4 apresenta a conclusão.

2 O Problema de Programação de Jogos Abordado

O problema de programação de jogos de competições esportivas realizadas em dois turnos espelhados (PPJE) é o problema típico de alocação de jogos organizado pela Confederação Brasileira de Futebol (CBF). O problema é constituído da seguinte maneira: cada um dos n times participantes da competição joga duas vezes contra os demais, uma em sua sede e outra na sede do oponente, não necessariamente nessa ordem. No PPJE, os jogos são realizados em dois turnos completos (isto é, todos os times jogam em todas as rodadas), sendo que os jogos do segundo turno são os mesmos do primeiro turno, inclusive na mesma ordem, porém com mando de campo invertido. Assume-se que a cada time do torneio está associado um estádio, localizado em uma certa cidade, considerada sua sede, e que as distâncias entre as sedes dos times sejam conhecidas. Nesse problema, considera-se que cada time inicia o campeonato deslocando-se de sua sede se fizer a primeira partida fora de casa e a ela retornando ao final do campeonato, caso faça a última partida fora de casa. Quando um time jogar duas partidas consecutivas fora de casa, assume-se que ele sairá da cidade do primeiro oponente diretamente para a cidade do segundo, sem retornar à sua sede. A Tabela 1 ilustra uma escala de jogos de uma competição envolvendo seis times.

Tabela 1: Exemplo de uma escala de jogos do PPJE

		Rodadas	Jogos		
1º Turno		1	1 x 5	4 x 2	6 x 3
		2	3 x 2	1 x 6	5 x 4
		3	4 x 6	2 x 5	3 x 1
		4	5 x 6	2 x 1	4 x 3
		5	5 x 3	6 x 2	1 x 4
2º Turno		6	5 x 1	2 x 4	3 x 6
		7	2 x 3	6 x 1	4 x 5
		8	6 x 4	5 x 2	1 x 3
		9	6 x 5	1 x 2	3 x 4
		10	3 x 5	2 x 6	4 x 1

Na Tabela 1, as cinco primeiras rodadas correspondem aos jogos do primeiro turno e as cinco últimas correspondem aos jogos do segundo turno. Em cada célula há um confronto da forma $T_i \times T_j$, que significa que o time T_i joga na sua sede contra o time T_j . Por exemplo, na rodada 1 há a notação “1 x 5”, que expressa que o time 1 joga em sua sede contra o time 5. Observa-se, também, que a primeira rodada do segundo turno (rodada 6) contém os jogos inversos aos jogos da primeira rodada do primeiro turno (rodada 1), ou seja, a primeira rodada do segundo turno é o “espelho” da primeira rodada do primeiro turno. O mesmo ocorre para as demais rodadas do primeiro e segundo turno da tabela, caracterizando-se, assim, uma escala do PPJE.

São as seguintes as restrições consideradas no PPJE, válidas para os campeonatos da primeira divisão do Campeonato Brasileiro de Futebol em 2004 e 2005:

- Cada time joga somente uma vez por rodada;
- Dois times jogarão entre si duas vezes, uma no turno e a outra no retorno, alternando o mando de campo entre os mesmos;
- Nas duas primeiras rodadas de cada turno, cada time alternará seus jogos, sendo um em casa e o outro na casa do adversário;
- As duas últimas rodadas de cada turno terão a configuração inversa das duas

- primeiras rodadas de cada turno com relação ao mando de campo;
- e) Não poderá haver jogos entre times do mesmo estado na última rodada;
 - f) A diferença entre os jogos feitos em cada turno em casa e fora de casa de um time não pode ser maior que uma unidade;
 - g) Um time não pode jogar mais que duas vezes consecutivas dentro ou fora de casa.

O PPJE tem dois objetivos. O principal é minimizar a distância total viajada por todos os times durante o campeonato. O objetivo secundário é minimizar a diferença entre a distância do time que mais viajou e o que menos viajou no campeonato, balanceando desta forma, a distância viajada por cada time.

3 Metodologia

Neste capítulo é apresentada a metodologia proposta para gerar uma solução inicial aleatória para o problema de programação de jogos do Campeonato Brasileiro de Futebol. Na seção 3.1 mostra-se como uma solução para o PPJE é representada. Na seção 3.2 descreve-se como se uma solução do PPJE é avaliada e na seção 3.3 é apresentado o método proposto neste trabalho.

3.1 Representação de uma solução

A representação de uma solução adotada neste trabalho é a utilizada por Anagnostopoulos *et al.* (2003). Seja n o número de times envolvidos e $nr = 2n - 2$ o número de rodadas do campeonato. A solução nesta representação é uma matriz de n linhas por nr colunas. Cada linha referencia a um time T_i , cada coluna representa uma rodada R_k e cada célula (i, R_k) representa o oponente do time T_i na rodada R_k . Um sinal associado ao oponente denota o mando de campo, ou seja, caso o sinal seja positivo, significa que o time T_i irá jogar contra um oponente em sua própria sede. Caso o sinal seja negativo, significa que o time T_i irá jogar contra seu oponente na sede do time que está confrontando. A Tabela 2 exemplifica uma solução na representação de Anagnostopoulos *et al.* (2003), considerando 6 times e 10 rodadas.

Tabela 2: Exemplo de uma solução na representação de Anagnostopoulos *et al.* (2003)

Time \ Rodada	1	2	3	4	5	6	7	8	9	10
1	+ 6	- 5	+ 4	+ 3	- 2	- 4	- 3	+ 2	+ 5	- 6
2	+ 5	- 3	+ 6	+ 4	+ 1	- 6	- 4	- 1	+ 3	- 5
3	- 4	+ 2	+ 5	- 1	+ 6	- 5	+ 1	- 6	- 2	+ 4
4	+ 3	+ 6	- 1	- 2	- 5	+ 1	+ 2	+ 5	- 6	- 3
5	- 2	+ 1	- 3	- 6	+ 4	+ 3	+ 6	- 4	- 1	+ 2
6	- 1	- 4	- 2	+ 5	- 3	+ 2	- 5	+ 3	+ 4	+ 1

Na Tabela 2, observa-se, por exemplo, que a notação +5, encontrada na primeira rodada do time 2, indica que o time 2 joga em sua sede contra o time 5, enquanto que na rodada 4 do time 3, encontra-se a notação -1, cujo significado é que o time 3 joga fora de sua sede (no caso, na casa do oponente) contra o time 1.

Nesta representação, pode-se verificar a trajetória percorrida por um time durante o campeonato. Como exemplo, será mostrada a trajetória do time 6 na solução da Tabela 2. O time 6 inicia o campeonato dentro de sua sede e, na primeira rodada, desloca-se para a sede do time 1. Em seguida, desloca-se para a sede do time 4 (na segunda rodada) sem retornar à sua própria sede. Na rodada 3, ele segue da sede do time 4 para a sede do time 2. Na quarta rodada, ele retorna à sua sede (pois o jogo será 6×5, na sede do time 6) e desloca-se para a sede do time 3, na quinta rodada e assim sucessivamente até a 10ª rodada, onde encerrará o campeonato em sua sede. Isso não ocorre com o time 2, pois seu último confronto será na sede do time 5. Neste caso, é necessário que o time 2 retorne à sua sede após a décima rodada, para que assim possa encerrar o campeonato. As trajetórias dos times 2 e 6 são descritas a seguir, onde a notação $T_i \xrightarrow{r} T_j$ significa que o time T_i desloca-se para a sede do time T_j , na

rodada r ; e a notação $T_i \mapsto T_j$ significa que o time T_j retorna da sede do time T_i à sua sede, no final do campeonato:

Trajectoria do time 6: $6 \xrightarrow{1} 1 \xrightarrow{2} 4 \xrightarrow{3} 2 \xrightarrow{4} 6 \xrightarrow{5} 3 \xrightarrow{6} 6 \xrightarrow{7} 5 \xrightarrow{8} 6$

Trajectoria do time 2: $2 \xrightarrow{2} 3 \xrightarrow{3} 2 \xrightarrow{6} 6 \xrightarrow{7} 4 \xrightarrow{8} 1 \xrightarrow{9} 2 \xrightarrow{10} 5 \mapsto 2$

3.2 Função de avaliação

Uma solução s é avaliada pela função f apresentada pela fórmula (1). As duas primeiras componentes dessa função representam os objetivos propriamente ditos e a terceira penaliza o não atendimento às restrições do problema.

$$f(s) = \sum_{i \in T} \text{custo}(i) + \text{dif}(s) + \sum_{j \in C} w_j \times \text{inv}_j \quad (1)$$

em que $f(s)$: função de avaliação
 T : conjunto dos times participantes da competição;
 C : conjunto de restrições descritas no capítulo 2;
 $\text{custo}(i)$: custo de um time $i \in T$, conforme definido mais adiante;
 $\text{dif}(s)$: diferença entre o custo máximo e o custo mínimo dos times, isto é:
 $\text{dif}(s) = \max \{ \text{custo}(i), i \in T \} - \min \{ \text{custo}(i), i \in T \}$
 w_j : penalidade por desrespeitar a restrição $j \in C$;
 inv_j : número de vezes que a restrição $j \in C$ está sendo desrespeitada.

O custo de um time $i \in T$ é definido da seguinte forma. Seja a solução s apresentada na Tabela 2 e d_{ij} a distância entre as sedes dos times i e j e seja $\text{custo}(i)$ o custo de deslocamento efetuado pelo time i em toda a competição, considerando que no seu início cada time sai da sua sede e à ela retorna ao final da competição. Assim, por exemplo, o custo de deslocamento do time 1 na solução da Tabela 2, é calculado pela fórmula (2).

$$\text{custo}(1) = d_{15} + d_{51} + d_{12} + d_{24} + d_{43} + d_{31} + d_{16} + d_{61} \quad (2)$$

3.3 Geração de uma solução inicial

Uma solução inicial é gerada por um procedimento de duas fases, ambas baseadas em *backtracking*. Na primeira, é feita a alocação das duas primeiras e duas últimas rodadas do primeiro turno. Na segunda fase são geradas as rodadas intermediárias desse turno. A seguir, gera-se o segundo turno mantendo-se a mesma seqüência de jogos do primeiro turno, porém com o mando de campo invertido. Com esse procedimento é possível gerar uma solução inicial satisfazendo as restrições (c) e (d) descritas no capítulo 2. Caso não houvesse esse tratamento especial, essas restrições seriam difíceis de serem contempladas por um método de refinamento que utilizasse os movimentos propostos por Anagnostopoulos *et al.* (2003), os quais não são fortes o suficiente para eliminar, a partir de qualquer solução inicial, as inviabilidades decorrentes do não atendimento a essas restrições.

3.4.1. Primeira fase

Nessa fase, utiliza-se um algoritmo baseado em *backtracking* para gerar quatro rodadas, cujo objetivo é satisfazer às restrições (c) e (d) descritas no capítulo 2. Essas rodadas formarão as duas primeiras e as duas últimas rodadas do primeiro turno.

Inicialmente, gera-se uma matriz denominada “matriz base”, cuja composição é, inicialmente, de uma rodada com seus respectivos mandos de campo definidos aleatoriamente. Em seguida, gera-se uma matriz de domínios, constituída dos oponentes possíveis para um determinado time, segundo as configurações de mando de campo e os oponentes definidos na matriz base. O método baseado em *backtracking* é iniciado escolhendo-se aleatoriamente um oponente do primeiro time. Definido esse oponente, atualiza-se a matriz de domínios para que um time que já foi alocado não jogue mais contra um outro time na mesma rodada. Nas próximas iterações, o método consiste em escolher o oponente do próximo time e atualizar a matriz de domínios. Caso o domínio do próximo time seja vazio, deve-se fazer um processo de *backtracking*, que consiste em desalocar o jogo do time anterior, atualizar a matriz de domínios desconsiderando o oponente desalocado e escolher um outro oponente para esse time. Caso não haja oponentes, continua-se o processo de *backtracking* até que se encontre um outro oponente para o time corrente. Ao completar uma rodada, adiciona-se essa rodada à base e o processo de *backtracking* é forçado até que se atinja o primeiro time. Assim, uma nova rodada será iniciada a partir do próximo oponente do primeiro time. O método é encerrado após a formação das quatro rodadas. Se o domínio do primeiro time não possuir mais nenhum oponente e as quatro rodadas ainda não tiverem sido completadas, reinicia-se esta fase.

Ao final da primeira fase, duas das quatro rodadas irão formar a primeira e a segunda rodada, sendo que a segunda rodada terá a configuração de mando de campo inversa à da primeira rodada. As duas outras rodadas irão formar a penúltima e última rodada do primeiro turno, sendo que a configuração de mando de campo dessas duas últimas será inversa à configuração de mando de campo das duas primeiras.

Para exemplificar esse procedimento, considere uma competição envolvendo 8 times. A matriz base e a matriz de domínios para este caso é mostrada, respectivamente, nas Tabelas 3 e 4.

Tabela 3: Matriz base inicial

Times	MC	Rodadas			
1	+	4			
2	-	5			
3	+	6			
4	-	1			
5	+	2			
6	-	3			
7	+	8			
8	-	7			

Tabela 4: Matriz de domínios

Times	Domínios			
1	2	6	8	
2	1	3	7	
3	2	4	8	
4	3	5	7	
5	4	6	8	
6	1	5	7	
7	2	4	6	
8	1	3	5	

Na Tabela 3, a coluna *Times* representam os times, enquanto a coluna *MC* representa o mando de campo segundo a representação de Anagnostopoulos *et al.* (2003). Cada célula da coluna *Rodadas* representa um oponente do respectivo time.

O método baseado em *backtracking* é iniciado pela escolha aleatória de um oponente para o time 1, no caso, o time 2. Essa iteração é ilustrada na Tabela 5.

Tabela 5: 1ª Iteração

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5				1	3	7
3	+	6				2	4	8
4	-	1				3	5	7
5	+	2				4	6	8
6	-	3				1	5	7
7	+	8				2	4	6
8	-	7				1	3	5

A Tabela 6 apresenta a matriz de domínios atualizada.

Tabela 6: Matriz de domínios atualizada para a 1ª iteração

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5				1		
3	+	6				4	8	
4	-	1				3	5	7
5	+	2				4	6	8
6	-	3				5	7	
7	+	8				4	6	
8	-	7				3	5	

Nas próximas iterações, o método consiste em escolher o oponente do próximo time e atualizar a matriz de domínios. As cinco iterações seguintes são apresentadas nas Tabelas 7, 8, 9, 10 e 11.

Tabela 7: 2ª Iteração – Alocação do oponente do time 2

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6				4	8	
4	-	1				3	5	7
5	+	2				4	6	8
6	-	3				5	7	
7	+	8				4	6	
8	-	7				3	5	

Tabela 8: 3ª Iteração – Alocação do oponente do time 3

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1				3		
5	+	2				6	8	
6	-	3				5	7	
7	+	8				6		
8	-	7				5		

Tabela 9: 4ª Iteração – Alocação do oponente do time 4

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2				6	8	
6	-	3				5	7	
7	+	8				6		
8	-	7				5		

Tabela 10: 5ª Iteração – Alocação do oponente do time 5

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2	6			8		
6	-	3				5		
7	+	8						
8	-	7						

Tabela 11: 6ª Iteração – Alocação do oponente do time 6

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2	6			8		
6	-	3	5					
7	+	8						
8	-	7						

Observando-se a Tabela 11, pode-se verificar que na próxima iteração (7ª) não existe nenhum oponente para jogar contra o time 7, ou seja, o domínio do time 7 é vazio. Nesse caso, deve-se fazer o processo de *backtracking*. As duas iterações seguintes são mostradas nas Tabelas 12 e 13.

Tabela 12: 7ª Iteração – *Backtracking* 1

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2	6			8		
6	-	3						
7	+	8						
8	-	7						

Tabela 13: 8ª Iteração – *Backtracking 2*

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2	8					
6	-	3				7		
7	+	8				6		
8	-	7				5		

Observa-se na Tabela 13 que o time 5 ainda pode jogar contra o time 8. Assim, define-se o time 8 para jogar contra o time 5 e continua-se alocando os oponentes para os times restantes. A Tabela 14 mostra o resultado da 11ª iteração:

Tabela 14: 11ª Iteração – Alocação do oponente do time 8

Times	MC	Rodadas				Domínios		
1	+	4	2			6	8	
2	-	5	1					
3	+	6	4			8		
4	-	1	3					
5	+	2	8					
6	-	3	7					
7	+	8	6					
8	-	7	5					

Como na 11ª iteração foi possível construir uma rodada, o método finaliza a formulação da rodada corrente e inicia a próxima etapa. Essa etapa consiste em elaborar a 3ª rodada escolhendo, aleatoriamente, um time pertencente ao domínio corrente do time 1 e atualizando a matriz de domínios. A Tabela 15 mostra a primeira iteração para a construção da terceira rodada.

Tabela 15: 1ª Iteração – Alocação do oponente do time 1 para a terceira rodada

Times	MC	Rodadas				Domínios		
1	+	4	2	6		8		
2	-	5	1			3	7	
3	+	6	4			2	4	8
4	-	1	3			3	5	7
5	+	2	8			4	8	
6	-	3	7			5	7	
7	+	8	6			2	4	
8	-	7	5			3	5	

Assim, continua-se o processo sucessivamente até que sejam obtidas as quatro rodadas. Caso as quatro rodadas não possam ser obtidas, reinicia-se o método gerando-se uma nova rodada inicial aleatória, com seus respectivos mandos de campos também definidos aleatoriamente. A Tabela 16 mostra as rodadas obtidas pela primeira fase do método de geração da solução inicial.

Tabela 16: Rodadas geradas pela primeira fase do método

Times	MC	Rodadas			
1	+	4	2	6	8
2	-	5	1	7	3
3	+	6	4	8	2
4	-	1	3	5	7
5	+	2	8	4	6
6	-	3	7	1	5
7	+	8	6	2	4
8	-	7	5	3	1

Essas rodadas irão compor as duas primeiras e duas últimas rodadas do primeiro turno, sendo que a primeira e a última rodada terão a mesma configuração de mando de campo estabelecido em *MC* e a segunda e a penúltima rodada terão a configuração inversa de mando de campo de *MC*. A Tabela 17 mostra a escala obtida pela primeira fase do método de geração da solução inicial.

Tabela 17: Configuração da escala ao final da primeira fase

Time \ Rodada	1	2	3	4	5	6	7
1	+ 4	- 2				- 6	+ 8
2	- 5	+ 1				+ 7	- 3
3	+ 6	- 4				- 8	+ 2
4	- 1	+ 3				+ 5	- 7
5	+ 2	- 8				- 4	+ 6
6	- 3	+ 7				+ 1	- 5
7	+ 8	- 6				- 2	+ 4
8	- 7	+ 5				+ 3	- 1

3.4.2. Segunda fase

Na segunda fase do método de geração da solução inicial, geram-se as rodadas intermediárias do primeiro turno ainda não definidas, através de um processo de *backtracking* análogo ao procedimento utilizado na seção 3.4.1, diferindo-se nos aspectos apontados a seguir. A matriz base não possui nenhuma configuração de mando de campo pré-determinada. Conseqüentemente, a matriz de domínios é composta de todos os oponentes possíveis de um determinado time, excluindo-se aqueles alocados na primeira fase do método, e sem considerar um mando de campo específico. Para atribuir o mando de campo aos jogos das rodadas geradas nessa fase, utiliza-se um procedimento que procura alternar o mando de campo da seqüência de jogos de cada time, tal como em Ribeiro e Urrutia (2004).

A Tabela 18 mostra a escala resultante da segunda fase do método de geração da solução inicial.

Tabela 18: Configuração da escala ao final da segunda fase

Time \ Rodada	1	2	3	4	5	6	7
1	+ 4	- 2	+ 3	- 5	- 7	- 6	+ 8
2	- 5	+ 1	+ 4	+ 6	- 8	+ 7	- 3
3	+ 6	- 4	- 1	+ 7	+ 5	- 8	+ 2
4	- 1	+ 3	- 2	+ 8	- 6	+ 5	- 7
5	+ 2	- 8	+ 7	+ 1	- 3	- 4	+ 6
6	- 3	+ 7	- 8	- 2	+ 4	+ 1	- 5
7	+ 8	- 6	- 5	- 3	+ 1	- 2	+ 4
8	- 7	+ 5	+ 6	- 4	+ 2	+ 3	- 1

A seguir, essa escala é espelhada para formar o segundo turno do campeonato. A Tabela 19 mostra a solução inicial gerada pelo método de geração da solução inicial.

Tabela 19: Solução inicial

Time \ Rodada	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	+ 4	- 2	+ 3	- 5	- 7	- 6	+ 8	- 4	+ 2	- 3	+ 5	+ 7	+ 6	- 8
2	- 5	+ 1	+ 4	+ 6	- 8	+ 7	- 3	+ 5	- 1	- 4	- 6	+ 8	- 7	+ 3
3	+ 6	- 4	- 1	+ 7	+ 5	- 8	+ 2	- 6	+ 4	+ 1	- 7	- 5	+ 8	- 2
4	- 1	+ 3	- 2	+ 8	- 6	+ 5	- 7	+ 1	- 3	+ 2	- 8	+ 6	- 5	+ 7
5	+ 2	- 8	+ 7	+ 1	- 3	- 4	+ 6	- 2	+ 8	- 7	- 1	+ 3	+ 4	- 6
6	- 3	+ 7	- 8	- 2	+ 4	+ 1	- 5	+ 3	- 7	+ 8	+ 2	- 4	- 1	+ 5
7	+ 8	- 6	- 5	- 3	+ 1	- 2	+ 4	- 8	+ 6	+ 5	+ 3	- 1	+ 2	- 4
8	- 7	+ 5	+ 6	- 4	+ 2	+ 3	- 1	+ 7	- 5	- 6	+ 4	- 2	- 3	+ 1

4 Conclusão

Este trabalho apresenta uma metodologia de geração de uma solução inicial aleatória para o problema de programação de jogos da primeira divisão do Campeonato Brasileiro de Futebol.

Através desse método é possível gerar uma solução inicial que satisfaça as restrições (c) e (d) descritas no capítulo 2.

Referências Bibliográficas

ANAGNOSTOPOULOS, A.; MICHEL, L.; VAN HENTENRYCK; VERGADOS, Y. (2003) A Simulated Annealing Approach to the Traveling Tournament Problem. Proceedings of CP'AI'OR'03, Montreal, Canada.

RIBEIRO, C.; URRUTIA, S. (2004). Heuristics for the Mirrored Traveling Tournament Problem. *Proceedings of Practice and Theory of Automated Timetabling V*, PATAT, Pittsburgh, USA, p. 323-342.