PROGRAMAÇÃO DE JOGOS DA PRIMEIRA DIVISÃO DO CAMPEONATO BRASILEIRO DE FUTEBOL POR MEIO DA METAHEURÍSTICA *ITERATED LOCAL SEARCH*

Marcone Jamilson Freitas Souza
Universidade Federal de Ouro Preto, Departamento de Computação
35.400-000 Ouro Preto (MG)
marcone@iceb.ufop.br

Marcio Tadayuki Mine
Universidade Federal de Ouro Preto, Departamento de Computação
tadayuki@iceb.ufop.br

Matheus de Souza Alves Silva
Universidade Federal de Ouro Preto, Departamento de Computação
matheus@iceb.ufop.br

Luiz Satoru Ochi
Universidade Federal Fluminense, Instituto de Computação
Rua Passo da Pátria, 156 – Bloco E, 3º Andar, 24.210-240 Niterói (RJ)
satoru@dcc.ic.uff.br

Roberto Carlos Vieira Pontes
Optline Modelagem Matemática e Otimização
rcvpontes@yahoo.com.br

Resumo

Este trabalho aborda o problema de programação de jogos da primeira divisão do campeonato brasileiro de futebol. Este problema consiste em organizar uma tabela de jogos para os times participantes do campeonato. A competição é realizada em dois turnos, sendo os jogos do segundo turno realizados na mesma seqüência de confrontos do primeiro, porém com o mando de campo invertido. O objetivo principal é minimizar o deslocamento total efetuado pelos times envolvidos, respeitando-se um determinado conjunto de restrições, entre as quais: (a) um time não pode jogar mais que duas vezes consecutivas em sua sede ou fora dela; (b) cada time deve jogar duas vezes contra todos os demais, sendo que um desses jogos é realizado em sua própria sede e o outro na sede do adversário. Neste trabalho, o problema é abordado pela metaheurística *Iterated Local Search*, usando-se quatro tipos diferentes de movimentos para explorar o espaço de soluções viáveis do problema. A metodologia desenvolvida é testada com as instâncias reais de 2004 e 2005 relativas ao Campeonato Brasileiro de Futebol da primeira divisão e comparada com as tabelas geradas pela entidade organizadora do evento, a Confederação Brasileira de Futebol (CBF), bem como com um resultado apresentado na literatura para o campeonato de 2004. Mostra-se que a metodologia proposta é capaz de melhorar significativamente as tabelas aplicadas pela CBF, além de superar o resultado da literatura.

Palavras-Chaves: Programação de jogos de competições esportivas; Campeonato Brasileiro de Futebol; Metaheurísticas; *Iterated Local Search*.

Abstract

This work presents a metaheuristic for scheduling of the Brazilian Soccer Championship (BSC) games. This problem consists of organizing a timetable of the BSC games for a set of teams. This competition occurs in two rounds. We suppose that each team has it own stadium and for each pair of them, two games occur, one in each stadium. The main objective of this problem is to minimize the total traveled distance by the teams satisfying a set of constraints, including: (a) no more than two consecutive home or away games are allowed for any times; (b) each team must play with each other twice, one game at home and the other at the opponent´s home. In this work, the problem is solved by the Iterated Local Search metaheuristic using four types of moves in order to explore the feasible solution space. The developed methodology is tested with real instance of 2004-BSC and 2005-BSC and the obtained results indicate that the proposed algorithm outperforms the results from the literature.

Keywords: Traveling Tournament Problem; Brazilian Soccer Championship; Metaheuristics, Iterated Local Search.

1. INTRODUÇÃO

As competições esportivas despertam a atenção de um grande número de pessoas e, por isso, têm sido de grande interesse para a mídia, já que representam fontes geradoras de lucro. Por outro lado, a organização desses eventos envolve, em muitos casos, como por exemplo, o Campeonato Brasileiro de Futebol, somas fabulosas. Esses valores representam os gastos com os deslocamentos dos times participantes da competição, sendo de interesse, portanto, sua redução. Para a redução desses custos é de suma importância um bom escalonamento dos jogos. Além do aspecto financeiro, ressalta-se a importância de equilibrar, entre os times, a distância total percorrida por eles ao longo da competição, de maneira que o desempenho dos times não seja afetado de forma desigual pelo desgaste dos jogadores com os deslocamentos.

Na literatura, o problema de organizar uma tabela de jogos de uma competição esportiva realizada em vários locais ao longo de um determinado período é conhecido como Problema de Programação de Jogos (PPJ) ou *Traveling Tournament Problem* (TTP) (veja Dorigo *et al*. (2001)). O principal objetivo do problema é minimizar o deslocamento total efetuado pelos times envolvidos, respeitando-se um determinado conjunto de restrições. No caso mais comum, as rodadas são completas, isto é, todos os times jogam na rodada, e cada time joga contra todos os demais duas vezes, sendo uma na sua sede e outra na sede do adversário. O problema pode ser espelhado ou não espelhado. Será espelhado quando os jogos do segundo turno forem realizados na mesma seqüência dos jogos do primeiro turno, porém com mando de campo invertido. No problema não espelhado impede-se apenas que os dois confrontos entre dois times não sejam consecutivos. Para ambos os casos, há um conjunto de problemas-teste na literatura, introduzidos inicialmente por Dorigo *et al*. (2001) e Easton *et al*. (2003), disponíveis em http://www.mat.gsia.cmu.edu/TOURN/, considerando a restrição adicional de que cada time não pode jogar mais do que três jogos consecutivos dentro ou fora de casa.

A alocação ótima dos jogos é uma tarefa complexa, pois se enquadra na classe de problemas NP-difíceis (veja Easton *et al.* (2003)) sendo, portanto, um problema altamente combinatório. De acordo com Concílio e Zuben (2002), para n times confrontando-se entre si em turnos completos, o número de tabelas (soluções) possíveis é dado pela fórmula (1).

$$(n-1)!(n-3)!(n-5)!...(n-(n-1))!\times 2^{(n-1)\times\frac{n}{2}} \quad (1)$$

Para exemplificar a magnitude do espaço de soluções do problema, para uma competição com 20 times participantes há 2,9062 x 10^{130} tabelas possíveis. Dessa forma, a resolução desse problema exclusivamente por métodos exatos, está limitada, via de regra, a problemas de pequenas dimensões. Para dimensões mais elevadas, a abordagem mais comum é através de técnicas aproximadas ou heurísticas. . Essas técnicas apresentam como grande vantagem a sua flexibilidade para tratar de muitos problemas reais onde dados e/ou parâmetros são atualizados dinamicamente. A principal desvantagem dessas técnicas é a impossibilidade de garantir a otimalidade da sua solução. Entretanto, com o aparecimento das chamadas metaheurísticas, tais como Busca Tabu (Glover e Laguna, 1997), *Simulated Annealing* (Kirkpatrick *et al*., 1983), GRASP (Feo e Resende, 1995), Método de Pesquisa em Vizinhança Variável (Mladenovic e Hansen, 1997) e Algoritmos Genéticos (Goldberg, 1989), que possuem mecanismos para tentar escapar de ótimos locais ainda distantes de um ótimo global, muito progresso se tem conseguido com o objetivo de obter soluções aproximadas de boa qualidade.

Costa (1995) desenvolveu um método híbrido combinando as técnicas Algoritmos Genéticos e Busca Tabu para resolver um problema de escalonamento de jogos da Liga Americana de Hóquei. Noções de Inteligência Artificial são introduzidas para direcionar o processo de evolução natural. A fase de mutação do Algoritmo Genético é trocada por uma busca no espaço de solução comandada pelo método Busca Tabu. Ao invés de uma mutação aleatória mudar os componentes, cada indivíduo é submetido a um processo de otimização separado antes de interagir novamente com outros membros da população.

Concílio e Zuben (2002) abordam o problema da montagem da tabela de jogos do Campeonato Paulista de Futebol de 1997, que envolveu 16 times, utilizando programação evolutiva. Essa competição foi realizada em turno único, com rodadas cheias.

Crauwels e Van Oudheusden (2003) desenvolveram um procedimento baseado em Colônia de Formigas para tratar o TTP não espelhado. Nesse procedimento, cada time é representado por uma formiga, a qual deixa uma certa quantidade de “feromônio” sempre que completa uma rota, isto é, sempre que termina todos os seus jogos. A quantidade de feromônio deixada em um arco da rota é inversamente proporcional à distância percorrida. Arcos com maior quantidade de feromônio são estimulados a serem freqüentados por outras formigas. Após uma solução viável ser obtida, é realizado um procedimento de busca local que utiliza duas estruturas de vizinhança, aplicadas na forma do Método VND (Mladenovic e Hansen, 1997).

Anagnostopoulos *et al*. (2003) implementaram a metaheurística *Simulated Annealing* (Kirkpatrick *et al*., 1983) com um mecanismo de reaquecimento da temperatura, cujo propósito é escapar de ótimos locais a baixas temperaturas. Implementaram, também, uma estratégia de oscilação, que consiste em variar o peso dado ao não atendimento às restrições do problema, alternando, dessa forma, a análise do espaço de soluções viáveis e inviáveis. Com essa metodologia, os autores obtiveram quase todos os melhores resultados da literatura para as instâncias do TTP não espelhado.

Biajoli *et al*. (2003) abordam com *Simulated Annealing* a primeira divisão do Campeonato Brasileiro de Futebol de 2002. Essa competição envolveu 26 times e foi realizada em turno único em 29 rodadas. O método proposto melhorou a solução adotada pela entidade organizadora do evento em 11,4%, tendo como referência o deslocamento total dos times.

Biajoli (2003) estendeu o algoritmo anterior para resolver o problema relativo à primeira divisão do Campeonato Brasileiro de Futebol de 2003, competição realizada em dois turnos completos e envolvendo 24 clubes. O algoritmo desenvolvido utiliza um procedimento de reaquecimento para escapar de mínimos locais em baixas temperaturas. Os resultados obtidos foram 9,6% melhores do que aqueles gerados manualmente pela Confederação Brasileira de Futebol (CBF).

Biajoli *et al*. (2004) trataram a primeira divisão do Campeonato Brasileiro de Futebol de 2004 também com o método *Simulated Annealing*. Após a execução do método é aplicado um procedimento de refinamento na solução resultante. Trata-se de uma busca local que consiste em substituir cada time *i* por um time *j* e vice-versa para todos os jogos envolvendo os times *i* e *j*. A única restrição analisada nesse último processo é a restrição de jogos entre times do mesmo estado da federação na última rodada. Em relação à escala feita manualmente pela CBF, o método provou sua eficiência reduzindo em 12,8% a distância total viajada pelos times e melhorou em 38,4% o balanceamento da distância viajada entre eles.

Ribeiro e Urrutia (2004a) desenvolveram um método híbrido utilizando as metodologias GRASP (Feo e Resende, 1995) e o método ILS (Lourenço *et al*., 2002) para resolver os problemas-teste da literatura relativos ao TTP espelhado e não espelhado, bem como uma instância bastante adaptada do Campeonato Brasileiro de Futebol de 2003. Para explorar o espaço de soluções do problema foram considerados quatro movimentos: (a) *Home-away swap* (HAS), que consiste em inverter o mando de campo de um jogo; (b) *Team swap* (TS), que troca todos os jogos de dois times em todas as rodadas; (c) *Partial swap rounds* (PSR), que considera quatro times e permuta os dois jogos entre eles de uma rodada r_1 por dois outros jogos entre eles de uma outra rodada r_2; (d) *Game rotation* (GR), que consiste em forçar a alocação de um jogo a uma rodada, seguida de uma atualização determinística da solução através do movimento nomeado como *ejection chain move*. Na fase de construção GRASP é utilizado uma heurística constituída de três fases: Na primeira fase, uma escala é construída para o problema de turno único, com *n* times abstratos. Na segunda fase, os times reais, cuja distância entre suas sedes são pequenas, são preferencialmente associados aos times abstratos que possuem maior número de oponentes consecutivos. Na terceira fase, é estabelecido o local de realização de cada jogo. A fase de construção termina com um simples processo de busca local. Na fase de refinamento do GRASP é utilizada a metaheurística ILS, tendo como método de busca local o método VND (Mladenovic e Hansen, 1997), o qual utiliza os movimentos HAS, TS e PRS. As soluções ótimas locais são perturbadas com o movimento GR. Com esse método os autores detêm as melhores soluções da literatura para o TTP espelhado.

Várias outras metodologias são apresentadas na literatura para resolver problemas de programação de jogos. Problemas relacionados a competições de beisebol são abordados em Cain Jr. (1977) e Russel e Leung (1994); enquanto Nemhauser e Trick (1998), Campbell e Chen (1976), Bean e Birge (1980) e Henz (2001) tratam competições de basquetebol. Schreuder (1992) analisa aspectos combinatoriais na construção da tabela de jogos do Campeonato Alemão de Futebol. Schaefer (1999) e de Werra (1988) exploram o tema sem considerar uma aplicação específica.

Pelos resultados ilustrados na literatura afim, conclui-se que o escalonamento automatizado dos jogos em competições esportivas é, de grande importância, pois usualmente consegue obter reduções significativas nos custos envolvidos. Observa-se que as instâncias da literatura que retratam o problema espelhado, disponíveis em http://www.mat.gsia.cmu.edu/TOURN/, não consideram restrições comuns a competições esportivas latinas, como as brasileiras. Nesse sentido, este trabalho apresenta uma metodologia para resolver instâncias reais do problema de programação de jogos do Campeonato Brasileiro de Futebol. O método, denominado ILS-MRD, parte de uma solução inicial gerada por *backtracking*. A seguir, esta solução é refinada pela metaheurística *Iterated Local Search* (Lourenço *et al*., 2002), a qual considera, para a exploração do espaço de soluções, quatro tipos diferentes de movimentos.

Este trabalho está organizado como segue. Na seção 2 descreve-se o problema abordado. A seção 3 apresenta a metodologia abordada. A seção 4 apresenta os resultados encontrados e a seção 5 conclui o trabalho apontando para trabalhos futuros.

2. DESCRIÇÃO DO PROBLEMA 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 está associado um estádio, localizado em uma certa cidade, considerada sua sede, e que as distâncias entre as sedes dos times sejam conhecidas. Considera-se que cada time inicia o campeonato na sua sede, deslocando-se dela 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									
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1 x 5	3 x 2	4 x 6	5 x 6	5 x 3	5 x 1	2 x 3	6 x 4	6 x 5	3 x 5
4 x 2	1 x 6	2 x 5	2 x 1	6 x 2	2 x 4	6 x 1	5 x 2	1 x 2	2 x 6
6 x 3	5 x 4	3 x 1	4 x 3	1 x 4	3 x 6	4 x 5	1 x 3	3 x 4	4 x 1
1º Turno					**2º Turno**				

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 × 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 segundo 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:

a) Cada time joga somente uma vez por rodada;

b) Dois times jogarão entre si duas vezes, uma no turno e a outra no returno, alternando o mando de campo entre os mesmos;

c) Nas duas primeiras rodadas de cada turno, cada time alternará seus jogos, sendo um em casa e o outro na casa do adversário;

d) 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 objetivo principal do PPJE 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, de forma a balancear a distância viajada por cada time.

Observa-se que no modelo de Ribeiro e Urrutia (2004a, 2004b), os autores trabalham apenas com o objetivo principal. A restrição (g) é também diferente, pois é permitido que cada time jogue até três vezes consecutivas dentro ou fora de casa. Além disso, os autores tratam o campeonato de 2003, cuja restrição (d) é diferente. Esses autores, entretanto, não mencionam as restrições consideradas. De qualquer forma, tratam um problema bastante diferente do real.

3. METODOLOGIA

3.1. REPRESENTAÇÃO DE UMA SOLUÇÃO

Adota-se a representação de uma solução de 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. O sinal positivo indica que o time T_i irá jogar contra um oponente em sua própria sede. Um sinal negativo indica, por sua vez, que o time T_i jogará contra seu oponente na sede do time que está confrontando. A Tabela 2 exemplifica uma solução, considerando 6 times e 10 rodadas.

Tabela 2: Exemplo de 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, a trajetória percorrida por um time durante o campeonato é determinada facilmente. Como exemplo, será mostrada a trajetória do time 6 na solução da Tabela 2, o qual 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 rodada 2) 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. A notação $T_i \xrightarrow{r} T_j$ indica que o time T_i desloca-se para a sede do time T_j na rodada r, e $T_i \mapsto T_j$ significa que o time T_j retorna da sede do time T_i à sua sede, no final do campeonato:

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

Trajetória 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. VIZINHANÇA DE UMA SOLUÇÃO

Para explorar o espaço de soluções do problema aplicam-se, neste trabalho, os movimentos *swap rounds*, *swap teams* e *swap homes*, utilizados por Anagnostopoulos *et al*. (2003), e mais um movimento, denominado *replace teams*, utilizado por Biajoli *et al*. (2004). As tabelas a seguir ilustram a geração de vizinhos por meio desses movimentos.

Considere a solução s apresentada pela Tabela 2. O movimento *swap rounds* (permutação de duas colunas) consiste em trocar todos os jogos de uma rodada R_i com os da rodada R_j. Efetuando-se este movimento na solução *s* para as rodadas 3 e 7, por exemplo, tem-se como resultado a solução apresentada na Tabela 3.

Tabela 3: Solução vizinha gerada pelo movimento *swap rounds*

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

O movimento *swap teams* (permutação de duas linhas) consiste em trocar todos os jogos entre dois times T_i e T_j, de todas as rodadas, com exceção daquelas em que eles se confrontam. Este movimento é realizado em duas partes: A primeira consiste em trocar os jogos entre os times T_i e T_j e a segunda, em restabelecer a viabilidade da tabela gerada. A Tabela 4 considera a primeira parte do movimento de troca entre os times 2 e 5 aplicado à solução *s* apresentada na Tabela 2.

Tabela 4: Solução gerada pelo movimento *swap teams* – parte 1

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	+ 1	- 3	- 6	+ 4	+ 3	+ 6	- 4	- 1	- 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	- 3	+ 6	+ 4	+ 1	- 6	- 4	- 1	+ 3	+ 2
6	- 1	- 4	- 2	+ 5	- 3	+ 2	- 5	+ 3	+ 4	+ 1

Como se observa, a simples troca dos jogos gera inconsistência na escala. Assim, a segunda parte do movimento efetua, de forma determinística, o restabelecimento da viabilidade da solução. A Tabela 5 mostra a solução produzida ao final do movimento *swap teams*.

Tabela 5: Solução vizinha gerada pelo movimento *swap teams* – parte 2

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

O movimento *swap homes* consiste em trocar o mando de campo dos dois confrontos envolvendo dois times. A Tabela 6 ilustra o resultado da aplicação desse movimento à solução *s* da Tabela 2, considerando os dois confrontos envolvendo os times 1 e 4.

Tabela 6: Solução vizinha gerada pelo movimento *swap homes*

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

O movimento *replace teams* consiste em substituir um determinado time T_i por outro time T_j e vice-versa. A Tabela 7 mostra o resultado da aplicação desse movimento à solução *s* da Tabela 2, considerando os dois confrontos envolvendo os times 1 e 2.

Tabela 7: Solução vizinha gerada pelo movimento *replace teams*

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

3.3. FUNÇÃO DE AVALIAÇÃO

Uma solução *s* é avaliada pela função *f* apresentada pela fórmula (2). As duas primeiras componentes dessa função representam os objetivos propriamente ditos e a terceira penaliza o não atendimento às restrições do problema.

$$f(s) = \sum_{i \in T} custo(i) + dif(s) + \sum_{j \in C} w_j \times inv_j \quad (2)$$

em que
- $f(s)$: função de avaliação
- T: conjunto dos times participantes da competição;
- C: conjunto de restrições descritas na seção 2;
- $custo(i)$: custo de um time $i \in T$, conforme definido mais adiante;
- $dif(s)$: diferença entre o custo máximo e o custo mínimo dos times, isto é: $dif(s) = \max\{custo(i),\ i \in T\} - \min\{custo(i),\ i \in T\}$
- 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 $custo(i)$ 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 (3).

$$custo(1) = d_{15} + d_{51} + d_{12} + d_{24} + d_{43} + d_{31} + d_{16} + d_{61} \quad (3)$$

3.4. 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) da seção 2. Caso não houvesse esse tratamento especial, essas restrições seriam difíceis de serem contempladas pelo método de refinamento, uma vez que os movimentos utilizados não são fortes o suficiente para eliminar, a partir de qualquer solução inicial, as inviabilidades decorrentes do não atendimento a essas restrições.

Detalhes desse procedimento podem ser encontrados em Silva *et al*. (2005).

A Tabela 8 mostra uma solução inicial gerada pelo método para uma competição envolvendo 8 times.

Tabela 8: Exemplo de uma solução inicial gerada pelo método de *backtracking*

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

Observa-se na Tabela 8 que as restrições (c) e (d) da seção 2 estão satisfeitas, isto é, nas duas primeiras rodadas de cada turno (rodadas 1 e 2 do primeiro turno e 8 e 9 do segundo turno) o mando de campo é alternado e nas duas últimas rodadas de cada turno (rodadas 6 e 7 do primeiro turno e 13 e 14 do segundo turno), o mando de campo é invertido em relação às duas primeiras rodadas respectivas de cada turno.

3.5. *ITERATED LOCAL SEARCH* APLICADA AO PPJE

Para refinar uma solução gerada pelo método descrito na seção 3.4, foi proposto um método híbrido, ILS-MRD, composto pela metaheurística Busca Local Iterada (*Iterated Local Search*, ILS) (Lourenço *et al*., 2002) e pelo Método Randômico de Descida (MRD), sendo este usado como um mecanismo de busca local para o ILS. O pseudocódigo do procedimento ILS adaptado ao PPJE é apresentado na Figura 1.

procedimento ILS-MRD
    $s_0 \leftarrow$ *SolucaoInicialAleatoria*
    $s \leftarrow MRD(s_0, IterMRD)$
    kp $\leftarrow kp_0$
    iter $\leftarrow 0$
    enquanto $(kp < kp_{max})$
        enquanto $(iter \leftarrow melhorIter < iter_{max})$
            $iter \leftarrow iter + 1$
            $s' \leftarrow$ *pertubação*(s, kp)
            $s'' \leftarrow MRD(s', IterMRD)$
            se $(f(s'') < f(s))$ **faça**
                $s \leftarrow s''$
                $melhorIter \leftarrow iter$
                kp $\leftarrow kp_0$
            fim-se
        fim-enquanto
        $kp \leftarrow kp + delta$
    fim-enquanto
    retorne s

Figura 1: Algoritmo ILS-MRD

Esse algoritmo parte de uma solução inicial aleatória, s_0, gerada por meio do método descrito na seção 3.4. Em seguida, aplica-se um mecanismo de busca local, o Método Randômico de Descida (MRD). Nesse método escolhe-se aleatoriamente um tipo de movimento (dentre os quatro apresentados na seção 3.2) e, a partir dele, é gerado um vizinho aleatório. Se esse vizinho for melhor que a solução corrente, ele passa a ser a nova solução corrente. Caso contrário, outro vizinho é gerado. O método MRD pára quando um número máximo de iterações sem melhora, no caso, *IterMRD*, for atingido. A cada iteração do método ILS-MRD, gera-se uma perturbação na solução corrente. Essa perturbação consiste em realizar k movimentos em uma estrutura de vizinhança escolhida aleatoriamente, sendo k um número arbitrário compreendido entre um e kp. A essa nova solução vizinha s' , aplica-se o MRD, gerando uma solução vizinha ótima local, s''. Se a solução s'' for melhor que a solução s corrente, então s'' é aceita como a nova solução corrente e reinicia-se o valor de kp. Essa repetição é executada até que $iter_{max}$ iterações sem melhora sejam realizadas. Quando isso ocorrer, incrementa-se o grau de perturbação, kp, em um fator *delta*, até que kp atinja seu valor máximo (kp_{max}).

4. RESULTADOS COMPUTACIONAIS

Apresentam-se, nessa seção, os resultados obtidos aplicando-se o método ILS-MRD (*Iterated Local Search* e Método Randômico de Descida) para resolver o PPJE. O programa foi desenvolvido na linguagem C utilizando o ambiente Borland C++ Builder, versão 5.0. Foi utilizado um computador ATLHON XP 2,0 GHz com 256 MB de memória RAM, rodando no sistema operacional Windows XP.

Para validar os métodos, foram utilizadas duas instâncias encontradas na literatura, disponíveis em http://www.decom.ufop.br/prof/marcone/projects/ttp/bssp.html, que se referem à primeira divisão do Campeonato Brasileiro de Futebol de 2004 e 2005. Observa-se que esta competição assumiu os moldes atuais a partir de 2004. Em 2003 a regra para a restrição (d), apontada na seção 2, era diferente. Antes de 2003, o campeonato era realizado em turno único e os times não necessariamente jogavam em todas as rodadas. Em 2004 a competição reuniu 24 times de 9 estados da federação. Em 2005, quatro desses times desceram para a segunda divisão do campeonato e dois novos times ascenderam à primeira divisão, resultando em 22 times participantes do campeonato de 2005, representando 10 estados da federação. Para cada instância foram realizadas 100 execuções, cada qual partindo de uma semente diferente de números aleatórios. Os parâmetros adotados no método ILS-MRD foram: $kp_0 = 1$, $kp_{max} = 5$, $iter_{max} = 350$, $delta = 2$, $IterMRD = 1000$.

Na Tabela 9, *Melhor Valor* é o melhor valor encontrado na literatura até então e *Melhor*, *Média* e *Desvio* são, respectivamente, o melhor valor encontrado pelo método ILS-MRD, a média em 100 execuções e o desvio percentual, obtido por meio da fórmula (4), e *TM* é o tempo médio, em segundos, de processamento do método em 100 execuções:

$$Desvio = 100 \times \frac{Média - MelhorValor}{MelhorValor} \tag{4}$$

Tabela 9: Resultados computacionais obtidos pelo ILS-MRD

Instâncias	*Melhor Valor*	**ILS-MRD**			
		Melhor	*Média*	*Desvio*	*TM*
bssp2004	842789[1]	806134	825089	-2,1	1772
bssp2005	909119[2]	743621	760350	-16,4	1303

[1] Biajoli *et al*. (2004); [2] CBF

A Tabela 10 compara, em relação à distância total percorrida (*DIST*), à diferença de distância entre o time que mais viajou e o que menos viajou (*DIF*) e ao valor da função f de avaliação (*FO*) descrita na seção 3.3, as tabelas produzidas pela Confederação Brasileira de Futebol (CBF), o método *Simulated Annealing* de Biajoli *et al*. (2004) e o método ILS-MRD.

Tabela 10: Comparação entre as metodologias de resolução do PPJE

Instâncias	**CBF**			**Biajoli *et al*. (2004)**			**ILS-MRD**		
	DIST	*DIF*	*FO*	*DIST*	*DIF*	*FO*	*DIST*	*DIF*	*FO*
bssp2004	905316	86610	991926	789480	53309	842789	754935	51199	806134
bssp2005	838464	70655	909119	-	-	-	696800	46821	743621

Pelos resultados mostrados nas tabelas 9 e 10, observa-se que o ILS-MRD é um método robusto e eficiente, pois, em média, as soluções obtidas pelo método foram 2,1% melhor que a melhor solução encontrada na literatura para o campeonato de 2004 e 16,4% melhor que a solução elaborada pela CBF para o campeonato de 2005. A melhor solução encontrada pelo método reduziu a distância total percorrida por todos os times durante o campeonato de 2004 em 4,4% em relação à melhor solução encontrada na literatura (Biajoli *et al*., 2004) e 16,6% em relação à solução obtida manualmente pela CBF. Para o campeonato de 2005, o método reduziu essa distância em 16,9% em relação à tabela elaborada pela CBF. Houve, também, uma redução de 4,0% em relação à solução de Biajoli *et al*. (2004) e 40,9% em relação à solução da CBF na diferença entre o time que mais viajou e o que menos viajou no campeonato. No campeonato de 2005, o método proposto reduziu essa diferença em 33,7% em relação à solução da CBF. Considerando o custo de transporte aéreo a R$0,70/Km e uma delegação de 20 pessoas por cada time, a economia que o método ILS-MRD proporcionaria em relação à solução adotada pela CBF é de mais de R$2 milhões para o campeonato de 2004 e quase R$2 milhões para o campeonato de 2005. Em termos de tempo computacional, o método também se mostrou competitivo, pois a instância bssp2004 foi resolvida em cerca de 30 minutos, enquanto o algoritmo de Biajoli *et al*. (2004) encontrou sua melhor solução em 45 minutos em um computador com processador AMD Athlon, 1,5 GHz e 256 MB de RAM.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho aborda o problema de programação de jogos da primeira divisão do Campeonato Brasileiro de Futebol de 2004 e 2005. Uma solução inicial é gerada por um método de duas fases, baseado em *backtracking*, cujo propósito é obter uma solução aleatória satisfazendo a duas das restrições do problema. Na fase de refinamento da solução foi utilizada uma adaptação da metaheurística *Iterated Local Search*, denominada ILS-MRD, que explora um subespaço de soluções ótimas locais através do Método Randômico de Descida. Esse método navega no espaço de soluções utilizando quatro tipos de movimento, a saber, *swap teams*, *swap rounds*, *swap homes* e *replace teams*. O método ILS-MRD faz uso de um mecanismo que consiste em aumentar o grau de perturbação de forma sistemática sempre que não há melhora na solução corrente em um determinado número de iterações, retornando-se ao grau mínimo de perturbação quando ocorre alguma melhora. Esse mecanismo teve, na prática, influência significativa para melhorar a qualidade da solução final.

Os resultados obtidos mostram que o ILS-MRD é um método robusto e eficiente, pois mesmo as soluções médias foram melhores que as existentes na literatura. Mostrou-se, ainda, que a metodologia proposta é capaz de reduzir significativamente os gastos com transporte.

O método ILS é um método flexível, pois permite facilmente a substituição de seus componentes, como, por exemplo, os mecanismos de perturbação e busca local. Segundo Lourenço *et al*. (2002), o sucesso do ILS está centrado no conjunto de amostragem de ótimos locais, juntamente com a escolha da busca local, das perturbações e do critério de aceitação. Logo, de acordo com esses autores, é ainda possível aperfeiçoar o método ILS-MRD, substituindo-se a heurística de busca local MRD por outras técnicas ou incorporando ao método outros procedimentos de busca. Uma possibilidade é o uso do procedimento Reconexão por Caminhos (*Path Relinking*), que consistiria em armazenar um conjunto de soluções elite ao longo da busca e, a partir de uma dada solução caminhar-se-ia no espaço de soluções considerando uma das soluções elite como solução guia. Outra possibilidade de melhoria é a incorporação de outros tipos de movimento na exploração do espaço de soluções, tais como *game rotation* e *partial swap rounds*, como em Ribeiro e Urrutia (2004).

6. AGRADECIMENTOS

Os autores agradecem à FAPEMIG, processo CEX-429/04, e à UFOP, pelo apoio ao desenvolvimento do trabalho.

7. REFERÊNCIAS BIBLIOGRÁFICAS

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

[2] BEAN, J. C.; BIRGE, J. R. (1980) Reducting Traveling Costs and Player Fatigue in National Basketball Association. Interfaces, v. 10, p. 98-102.

[3] BIAJOLI, F. L.; MINE, O. M.; CHAVES, A. A.; SOUZA, M. J. F. (2003) Escala de jogos de torneios esportivos: uma abordagem via Simulated Annealing. Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional, SBPO, Natal, v. 1, p. 1295-1306.

[4] BIAJOLI, F. L. (2003) Resolução do Problema de Programação de Jogos do Campeonato Brasileiro de Futebol. Relatório Técnico DECOM/04, Departamento de Computação, Universidade Federal de Ouro Preto, disponível em http:www.decom.ufop.br/prof/ marcone/Orientacoes/rt-decom-04-2004.pdf. Acesso em 22/04/2005.

[5] BIAJOLI, F. L.; MINE, O. M.; CHAVES, A. A.; SOUZA, M. J. F.; LUCENA, A.; CABRAL, L. A. F.; PONTES, R. C. V (2004) Scheduling the Brazilian Soccer Championship: A Simulated Annealing Approach. Proceedings of Practice and Theory of Automated Timetabling V, PATAT, Pittsburgh, USA, p. 433-436.

[6] CAIN Jr, W.O. (1977) A Computer Assisted Heuristic Approach Used to Schedule the Major League Baseball Clubs. In Ladany, S.P. and Machol, R.E. (eds), Optimal Strategies in Sports, North-Holland, Amsterdan, p. 32-41.

[7] CAMPBELL, R. T.; CHEN, D. S. (1976) A Minimum Distances Basketball Scheduling Problem. In Machol, R.E., Ladany, S.P. and Morrison, D.G. (eds), Management Science in Sports, North-Holland, Amsterdan, p. 15-25.

[8] CONCÍLIO, R.; ZUBEN, F. J. (2002) Uma Abordagem Evolutiva para Geração Automática de Turnos Completos em Torneios. Revista Controle & Automação, v. 13, n. 2, p. 105-122.

[9] COSTA, D. (1995) An Evolutionary Tabu Search Algorithm and the NHL Scheduling Problem. INFORS, v. 33, n. 3, p. 161-178.

[10] CRAUWELS, H.; VAN OUDHEUSDEN, D. (2003) Ant Colony Optimization and Local Improvement. Proceedings of CP'AI'OR'03, Montreal, Canada.

[11] de WERRA, D. (1988) Some Models of Graphs for Scheduling Sports Competitions, Discrete Applied Mathematics, v. 21, p. 47-65.

[12] DORIGO, M.; NEMHAUSER, G.L.; TRICK, M.A. (2001) The Traveling Tournament Problem: Description and Benchmarks. Lecture Notes in Computer Science, v. 2239, p. 580-585.

[13] EASTON, K.; NEMHAUSER, G. L.; TRICK, M. A. (2003) Solving the Traveling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach. Lecture Notes in Computer Science, v. 2740, p. 100-109.

[14] FEO, T. A.; RESENDE, M. G. C. (1995) Greedy randomized adaptive search procedures, Journal of Global Optimization, v. 6, p. 109-133.

[15] GLOVER, F. e LAGUNA, M. (1997) Tabu Search. Kluwer Academic Publishers, Norwell, MA.

[16] GOLDBERG, D. E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Berkeley.

[17] HENZ, M. (2001) Scheduling a Major College Basketball Conference: Revisited, Operations Research, v. 49, p. 1-12.

[18] KIRKPATRICK, S.; GELLAT, D. C.; VECCHI, M. P. (1983) Optimization by Simulated Annealing. Science, v. 220, p. 671-680.

[19] LOURENÇO, H. R., MARTIN, O., STÜTZLE, T. (2002) Iterated Local Search. In F. Glover and G. Kochenberger (eds), Handbook of Metaheuristics, p. 321 - 353, Kluwer Academic Publishers, Norwell, MA.

[20] MLADENOVIC, N.; HANSEN, P. (1997) A Variable Neighborhood Search. Computers and Operations Research, v. 24, p. 1097-1100.

[21] NEMHAUSER, G. L.; TRICK, M. A. (1998) Scheduling a Major College Basketball Conference. Operations Research, v. 46, n. 1, p. 1-8.

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

[23] RIBEIRO, C.; URRUTIA, S. (2004b). OR on the ball: Applications in sports scheduling management. OR/MS Today, v. 31, p. 50-54.

[24] RUSSELL, R. A.; LEUNG, J. M. (1994) Devising a cost effective schedule for a baseball league. Operations Research, v. 42, p. 614-625.

[25] SCHAEFER, A. (1999). Scheduling Sport Tournaments using Constraint Logic Programming. Constraints, v. 4, p. 43-65.

[26] SCHREUDER, J. A. M. (1992) Combinatorial Aspects of Construction of Competition Dutch Professional Football Leagues. Discrete Applied Mathematics, v. 35, p. 301-312.

[27] SILVA, M. S. A.; MINE, M. T.; SOUZA, M. J. F. (2005) Um método de geração de solução inicial, baseado em backtracking, para o problema de programação de jogos do Campeonato Brasileiro de Futebol. Relatório Técnico DECOM 10/2005, Departamento de Computação, Universidade Federal de Ouro Preto, 17 p. Disponível em http://www.decom.ufop.br/prof/marcone/projects/ttp/Publications/rt-decom-10-2005.pdf

[28] TRICK, M. A. (2001) A Schedule-Then-Break Approach to Sports Timetabling. Lecture Notes in Computer Science, v. 2079, p. 242-252.