

Uma Abordagem Evolutiva para o *Mirrored Traveling Tournament Problem*

Fabício Lacerda Biajoli
LAC/INPE
lacerda@lac.inpe.br

Luiz Antônio Nogueira Lorena
LAC/INPE
lorena@lac.inpe.br

Resumo

O *Traveling Tournament Problem* é um problema de otimização que abstrai algumas características de torneios esportivos, tendo como objetivo a minimização das distâncias percorridas pelos times no decorrer do torneio. Este trabalho aborda a versão espelhada do TTP, conhecida como *Mirrored Traveling Tournament Problem* (mTTP). Para tal propõe-se o uso conjunto das técnicas Algoritmos Genéticos e Simulated Annealing. A validação dos resultados será feita a partir de instâncias disponíveis na literatura.

Palavras-chave: mirrored traveling tournament problem, sports timetabling, algoritmo genético, simulated annealing.

1. Introdução

O escalonamento de jogos, ou programação de jogos, vem se tornando nos últimos anos uma das classes mais importantes de problemas combinatórios. Isto porque as ligas esportivas profissionais representam uma das maiores atividades econômicas ao redor do mundo e para muitas delas (tais como futebol, futsal, voleibol, basquetebol, etc), onde os jogos são disputados dois a dois e realizados em vários locais ao longo de um determinado período de tempo, há a necessidade de se fazer um escalonamento desses jogos.

Além da necessidade de se escalonar estes jogos alguns fatores fortalecem a aplicação de técnicas de otimização em problemas desta natureza, tais como: times e ligas não querem perder seus investimentos em jogadores e estrutura em consequência de escalonamentos mal feitos ou mal organizados; as ligas esportivas representam significantes fontes de renda das redes de rádio e televisão mundial; as escalas interferem diretamente no desempenho dos times participantes da competição; entre vários outros fatores. Do lado dos estudiosos da Pesquisa Operacional o principal atrativo é que as ligas esportivas geram problemas de otimização extremamente desafiadores.

O problema de geração de escalas para torneios esportivos é conhecido na literatura como *Traveling*

Tournament Problem (Trick et al., 2001). Basicamente a tarefa de gerar uma escala de jogos consiste em fazer com que todo time participante da competição confronte no mínimo uma vez todos os outros (condição que varia entre as competições) e que todos os times joguem em todas as rodadas com oponentes diferentes (seja em sua sede ou fora dela).

Problemas dessa natureza contêm em geral muitas restrições conflitantes que devem ser satisfeitas e diferentes objetivos a cumprir, como a minimização dos deslocamentos dos times durante o campeonato, realização de apenas uma partida por time e por dia, realização de determinados jogos em estádios e em datas pré-estabelecidas, número mínimo de partidas consecutivas realizadas na sede do time e fora dela, etc.

A geração de escalonamentos satisfatórios, respeitando essas condições e objetivos, é um problema muito difícil de ser resolvido. A dificuldade de solução desse problema é atribuída ao grande número de possibilidades a serem analisadas. Para exemplificar a magnitude do espaço de soluções, para uma competição com 20 participantes há $2,9062 \times 10^{130}$ combinações possíveis (Concílio & Zuben, 2002).

Neste trabalho é proposta a aplicação conjunta de técnicas evolutivas e busca local na resolução da versão espelhada do TTP, conhecida como *Mirrored Traveling Tournament Problem* (mTTP) (Ribeiro & Urrutia, 2004).

Este trabalho está organizado em 6 seções, sendo esta a primeira. A seção 2 apresenta a descrição do problema abordado. Na seção 3 é apresentada a metodologia adotada, com as estruturas de vizinhança e o algoritmo proposto. Os resultados computacionais serão apresentados na seção 4. E por último, nas seções 5 e 6, serão apresentadas as conclusões a respeito do trabalho desenvolvido e a referências bibliográficas pesquisadas, respectivamente.

2. O problema abordado

O *Traveling Tournament Problem* (TTP) é um problema de escalonamento de torneios esportivos proposto inicialmente por Trick et al., 2001, como uma abstração das características da MLB - *Major League Baseball* - dos Estados Unidos. É um problema com

fortes componentes de otimização, uma vez que apresenta inúmeras restrições que possuem uma dependência muito grande entre si, além dos objetivos distintos para se cumprir.

Uma escala de jogos para o TTP, onde o número de times é par (n), deve ser realizada de tal forma que todo time jogue com cada um dos outros exatamente duas vezes, sendo um jogo dentro de sua sede e o outro fora. Um jogo é representado por pares de oponentes. Exatamente $2(n-1)$ rodadas são requeridas para representar todos os jogos de uma escala. As distâncias entre as cidades sedes dos times são dadas por uma matriz D , $n \times n$, sendo que as distâncias entre as sedes dos times são simétricas, ou seja, a distância entre a sede de um time A para a sede de um time B é igual à distância da sede de B para a sede de A. Cada time inicia o torneio em casa e começa a viajar para cumprir seus jogos nas sedes escolhidas. Cada time retorna, se necessário, para sua sede no fim do torneio.

As restrições associadas ao TTP são:

1. Todo time deve jogar duas vezes com cada um dos outros: n times necessitam $2(n-1)$ rodadas;
2. Não mais do que três partidas consecutivas dentro de casa;
3. Não mais do que três partidas consecutivas fora de casa;
4. Não haver repetição de jogos com mesmos oponentes (por exemplo, o jogo A x B seguido imediatamente por B x A);
5. Minimizar as distâncias de viagem total dos times participantes;

O *Mirrored Traveling Tournament Problem* (mTTP) foi proposto por Ribeiro & Urrutia, 2004, e representa uma generalização do TTP. O mTTP apresenta em alguns aspectos similaridades à torneios de futebol muito populares na América Latina (como, por exemplo, o Campeonato Brasileiro de Futebol). Dentre estas similaridades pode-se destacar como sendo a principal o conceito de turnos de jogos. Um turno pode ser comparado a um torneio simples, onde cada time joga com cada um dos outros exatamente uma vez, necessitando $n-1$ rodadas para completar a escala parcial. Em torneios com turno e retorno cada time joga com cada um dos outros duas vezes, sendo um jogo em casa e outro fora de casa. O mTTP pode ser considerado um torneio simples em suas $n-1$ primeiras rodadas, chamado primeiro turno, seguido pelo mesmo torneio com os mandos de campo invertidos em relação às $n-1$ primeiras rodadas, chamado retorno, que é o espelho do turno em relação aos mandos de campo. Isso quer dizer que, se um time A jogou em casa contra um time B na primeira rodada do turno, ele deverá jogar com o mesmo time B na primeira rodada do retorno, porém dentro da sede do time B.

Assim como no TTP deve-se assumir que os times devem estar em suas cidades sedes no início o no fim do torneio. Finalmente, a distância de viagem dos times é o somatório das distâncias entre as sedes dos times alocados aos jogos de cada rodada do turno e do retorno.

3. Metodologia

3.1. Representação de uma solução

Uma solução, escala de jogos S , é representada por uma tabela (Figura 1) indicando os oponentes de cada time, onde cada linha corresponde a um time e cada coluna corresponde a uma rodada. O oponente do time T_i na rodada r_k é representado pelo valor absoluto do elemento (i, k) . Se (i, k) é positivo, o jogo é realizado na sede do time T_i , caso contrário na sede de seu oponente, alocado na posição (i, k) da tabela. O número de rodadas representadas é igual a $n-1$, onde n é o número de times. Este número de rodadas equivale apenas aos jogos do turno. Optou-se por não representar o retorno pelo fato deste ser o espelho do turno, ou seja, mesma escala, porém com os mandos de campos invertidos.

	Turno					Retorno				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-3	-4	5	-6	-2	3	4	-5	6	2
2	6	5	3	-4	1	-6	-5	-3	4	-1
3	1	6	-2	5	4	-1	-6	2	-5	-4
4	-5	1	6	2	-3	5	-1	-6	-2	3
5	4	-2	-1	-3	6	-4	2	1	3	-6
6	-2	-3	-4	1	-5	2	3	4	-1	5

Figura 1. Representação de uma solução.

3.2. Estruturas de vizinhança

A vizinhança de uma escala S é o conjunto de escalas que podem ser obtidas pela aplicação de um dos dois tipos de movimentos básicos, descritos abaixo, sempre aplicados na escala do turno, mas refletindo, conseqüentemente, na escala do retorno.

3.2.1. Troca de mandos de campo - TMC

Este movimento realiza a troca do mando de campo de um jogo entre dois times T_i e T_j , sorteados aleatoriamente, em uma rodada r_k também sorteada aleatoriamente. Em outras palavras, se o time T_i joga com o time T_j na sede de T_j , na rodada r_k , o movimento de troca de mando de campo faz com que o time T_i jogue em sua sede, conseqüentemente o time T_j deve se deslocar até a sede de T_i , na rodada r_k . Considere a escala S apresentada abaixo, bem como os times T_i e T_4 .

	Turno					Retorno				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-3	-4	5	-6	-2	3	4	-5	6	2
2	6	5	3	-4	1	-6	-5	-3	4	-1
3	1	6	-2	5	4	-1	-6	2	-5	-4
4	-5	1	6	2	-3	5	-1	-6	-2	3
5	4	-2	-1	-3	6	-4	2	1	3	-6
6	-2	-3	-4	1	-5	2	3	4	-1	5

Figura 2. Escala S antes do movimento TMC.

O movimento de troca de mandos de campo aplicado aos times T_1 e T_4 na rodada r_2 gera a escala S' apresentada abaixo.

	Turno					Retorno				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-3	4	5	-6	-2	3	-4	-5	6	2
2	6	5	3	-4	1	-6	-5	-3	4	-1
3	1	6	-2	5	4	-1	-6	2	-5	-4
4	-5	-1	6	2	-3	5	1	-6	-2	3
5	4	-2	-1	-3	6	-4	2	1	3	-6
6	-2	-3	-4	1	-5	2	3	4	-1	5

Figura 3. Escala S' gerada pelo movimento TMC.

3.2.2. Troca times - TT

Este movimento troca todos os oponentes de dois times, T_i e T_j , escolhidos aleatoriamente. Apenas os jogos onde os times escolhidos se confrontam não são alterados. Em contrapartida, outros dois oponentes de cada um dos outros times também serão trocados, para manter a viabilidade da escala.

	Turno					Retorno				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-3	-4	5	-6	-2	3	4	-5	6	2
2	6	5	3	-4	1	-6	-5	-3	4	-1
3	1	6	-2	5	4	-1	-6	2	-5	-4
4	-5	1	6	2	-3	5	-1	-6	-2	3
5	4	-2	-1	-3	6	-4	2	1	3	-6
6	-2	-3	-4	1	-5	2	3	4	-1	5

Figura 4. Escala S antes do movimento TT.

A aplicação do movimento troca times em T_3 e T_5 gera uma nova escala, S' , apresentada na figura abaixo.

	Turno					Retorno				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-5	-4	3	-6	-2	5	4	-3	6	2
2	6	3	5	-4	1	-6	-3	-5	4	-1
3	4	-2	-1	5	6	-4	2	1	-5	-6
4	-3	1	6	2	-5	3	-1	-6	-2	5
5	1	6	-2	-3	4	-1	-6	2	3	-4
6	-2	-5	-4	1	-3	2	5	4	-1	3

Figura 5. Escala S' gerada pelo movimento TT.

3.3. Algoritmo proposto

Neste trabalho é proposta e aplicação conjunta de programação evolutiva e busca local, amparada pela otimização baseada em restrições que tem por objetivo definir prioridade de atendimento nas restrições do problema. Tal prioridade é conseguida a partir da definição de penalidades, que são aplicadas nas ocorrências de violação das restrições. A cada restrição uma penalidade é atribuída, sendo que, quanto maior a prioridade de atendimento da restrição, maior o valor da penalidade.

Foi implementado um Algoritmo Genético (AG) (Holland, 1975 e Goldberg, 1989) que faz uso da metaheurística *Simulated Annealing* (SA) (Kirkpatrick et al., 1983 e Dowsland et al., 1993) para direcionar os novos indivíduos de cada geração a ótimos locais. A aplicação de busca local nos indivíduos pode ser relacionada com a combinação de aprendizado e evolução (efeito *Baldwin* - Whitley et al., 1994). O aprendizado é, em geral, uma busca pela solução viável mais próxima e as modificações ocorridas serão incorporadas pelo indivíduo. O uso da metaheurística SA deixa a etapa de busca local mais agressiva, resultando em indivíduos cada vez mais adaptados dentro da população.

3.3.1. Algoritmo Genético

Trata-se de uma metaheurística que se fundamenta em uma analogia com processos naturais de evolução, nos quais, dada uma população, os indivíduos com características genéticas melhores têm maiores chances de sobrevivência e de produzir filhos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer. Para um melhor detalhamento do método sugere-se Holland, 1975 e Goldberg, 1989.

Para a aplicação do Algoritmo Genético é proposta uma representação compacta dos cromossomos (indivíduos da população), que passam por um algoritmo de expansão de código, o qual decodifica estes cromossomos em escalas de jogos. Dentro desta representação cada gene está associado a um time. A figura 6 apresenta um exemplo da representação compacta adotada.

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

Figura 6. Exemplo de um cromossomo para um torneio com 6 times.

Uma vez definido o cromossomo, este é submetido ao algoritmo de expansão de código, que por sua vez é a aplicação do famoso Método do Polígono (Dinitz et al., 1995) associado a uma heurística gulosa para definição dos mandos de campo, para geração da escala de jogos com $n-1$ rodadas (n é o número de times). É interessante ressaltar que todas escalas geradas por esta estratégia representam soluções viáveis.

A seguir é apresentado o funcionamento básico do Método do Polígono, quando aplicado num vetor de tamanho par.

Considere um vetor V de tamanho n onde cada posição i do vetor está associada a um time. A execução do método começa com a definição de um time base que é escolhido aleatoriamente e depois posicionado na primeira posição do vetor. Os demais times são dispostos nas posições restantes, de índice $i = 2, \dots, n$. Em cada rodada $r_i = 1, \dots, n-1$ o time base confronta sempre o time da posição $i = 2$.

Os times das posições $i = 3, \dots, (n/2)+1$ confrontam os times das posições de índice igual a $n-i+3$. Uma vez definidos todos os jogos da rodada r_i , os times são rotacionados no sentido horário do vetor, ou seja, o times das posições $i = 3, \dots, n$ passam para a posição $i-1$ e o time da posição 2 passa para a posição n . A figura 7 apresenta a aplicação do método do polígono, dado um vetor de times, onde $n = 6$, sendo 3 o time base.

Rodada 1	3	2	5	1	6	4
Rodada 2	3	5	1	6	4	2
Rodada 3	3	1	6	4	2	5
Rodada 4	3	6	4	2	5	1
Rodada 5	3	4	2	5	1	6

Figura 7. Método do Polígono.

A definição da população inicial é realizada de forma aleatória, sendo que cada indivíduo é submetido a uma busca local, neste trabalho o método de descida RNA - Randômico Não Ascendente - utilizando apenas o movimento de troca de mando de campo.

Um mecanismo de reprodução, baseado em processos evolutivos, é aplicado sobre a população com o objetivo de explorar o espaço de busca e encontrar melhores soluções para o problema. O operador de recombinação implementado foi baseado no cruzamento *Block Order Crossover* (BOX) (Syswerda, 1989). Os indivíduos "pais" são combinados, através da cópia aleatória de blocos de ambos indivíduos, o que resulta na geração de um novo "filho", contendo carga genética dos dois pais.

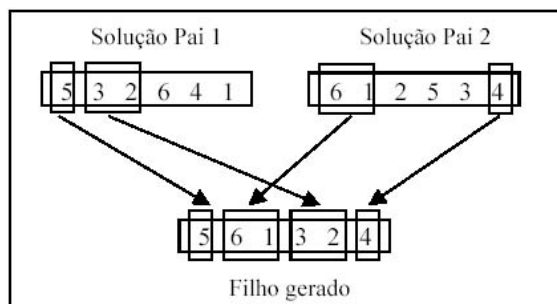


Figura 8. Exemplo do cruzamento BOX.

Como operador de mutação foi implementada uma descida *2-Troca*, que faz trocas entre dois genes de um cromossomo, guardando sempre a melhor troca realizada na solução corrente. Imediatamente após a execução dos operadores genéticos acima, o algoritmo de expansão de código é executado para geração da escala de jogos.

A figura 9 apresenta o pseudocódigo do AG implementado.

```

procedimento AG( )
1  GerarPopulaçãoInicial (P);
2  enquanto (i < nGerações) faça
3    enquanto (j < nIndividuos) faça
4      Selecionar Pais aleatoriamente em P;
5      filho = Recombinar(pais);
6      se(probabilidade de mutação aceita) então
7        mutação(filho);
8      fim-se
9      AvaliarVizinhança(filho);
10     Avaliar(filho);
11     InserirNaPopulação(filho, P);
12     Incrementar(j);
13   fim-enquanto
14   DefinirPopulacaoSobrevivente(P);
15   Incrementar(i);
16 fim-enquanto
fim-AG
    
```

Figura 9. Algoritmo Genético desenvolvido.

A metaheurística *Simulated Annealing* é aplicada a cada filho gerado ("AvaliarVizinhança(filho)", ver figura 9) tendo como objetivo o direcionamento dos novos indivíduos de cada geração a ótimos locais.

A idéia de se usar as metodologias citadas de forma conjunta se deve ao fato de que algoritmos evolutivos em sua forma tradicional encontram grandes dificuldades no tratamento de problemas de otimização com restrições, e também, por outro lado, a aplicação isolada de técnicas de otimização baseadas em restrições e busca local pode encontrar grandes dificuldades em virtude da existência de ótimos locais de baixa qualidade. Assim sendo, a aplicação conjunta das metodologias se justifica por dois pontos principais:

- O processo evolutivo vai poder atuar somente sobre a codificação compacta, junto a qual não há inviabilidade, fazendo com que sua eficácia não seja reduzida;
- A existência de uma população de candidatos à solução sendo evoluída, juntamente com a aplicação de um procedimento de busca local, aumenta significativamente as chances de se obter ótimos locais de boa qualidade.

4. Experimentos e resultados computacionais

O algoritmo proposto foi codificado em linguagem C++, rodando em compilador *Borland C++ Builder 5* e conduzidos em uma máquina *Pentium IV 3.0 GHz* com *512 de memória RAM*.

Com intuito de validar os resultados obtidos pela estratégia proposta foram utilizadas instâncias testes descritas em Trick et al., 2001, e utilizadas na forma espelhada em Ribeiro & Urrutia, 2004. Estas instâncias estão disponíveis para *download* no endereço <http://mat.gsia.cmu.edu/TOURN/>. A escolha dos parâmetros dos métodos utilizados foi realizada de forma empírica.

A tabela 1 apresenta os resultados obtidos. Na primeira coluna são apresentadas as instâncias testadas, na segunda os melhores resultados conhecidos, na terceira os resultados obtidos pelo algoritmo proposto, e nas duas últimas o *gap* e o tempo de processamento necessário para se obter cada uma das soluções, respectivamente.

Instâncias	Melhores	Obtidos	gap(%)	tempo(seg)
circ4	20	20	0 %	2
circ6	72	72	0 %	4
circ8	140	154	9,09 %	6
circ10	240	288	16,67 %	48
circ12	456	458	0,44 %	51
circ14	714	714	0 %	26
circ16	980	1014	3,35 %	264
circ18	1306	1370	4,67 %	604
circ20	1882	1890	0,42 %	568
nl4	8276	8276	0 %	2
nl6	26588	26588	0 %	3
nl8	41928	43774	4,22 %	25
nl10	58190	66264	12,18 %	130
nl12	120655	121070	0,34 %	29
nl14	208086	208086	0 %	140
nl16	279618	290442	3,73 %	72
br2003.24	503158	511256	1,58 %	938

Tabela 1. Resultados computacionais.

Os resultados apresentados na tabela acima demonstram que a metodologia proposta pode ser utilizada de forma competitiva, pois mesmo utilizando apenas dois movimentos (sendo estes os mais simples entre os já conhecidos) foi possível obter valores bem próximos aos melhores resultados conhecidos na literatura, chegando a zerar o *gap* em seis das dezessete instâncias e ficando bem próximo de zerá-lo em outras oito. Por outro lado, para as instâncias *circ8*, *circ10* e *nl10* os resultados obtidos não foram tão bons, demonstrando a necessidade de aprimoramentos no algoritmo, principalmente no que diz respeito às estruturas de vizinhança. Este fato é facilmente justificado. As escalas geradas nesta implementação se encontram todas dentro de um "molde", imposto pela estratégia utilizada na geração das soluções iniciais,

restringindo o espaço de busca. Esta questão pode ser tratada através da implementação de movimentos que gerem vizinhos fora do molde imposto, como, por exemplo, movimentos que fazem troca de jogos, troca parcial de rodadas, etc. Com a implementação destes movimentos o espaço de busca aumenta de forma considerável, tornando possível a obtenção de resultados de melhor qualidade.

5. Conclusões

Ligas, confederações, organizações esportivas estão recebendo atenção considerável nos últimos anos, uma vez que essas atividades esportivas envolvem grande quantidade de investimentos em infra-estrutura, jogadores, redes de televisão e rádio. Além disso, problemas de otimização relacionados a escalonamento esportivo geram desafios cada vez mais almejados e difíceis de se resolver.

Neste trabalho foi tratado de forma inédita o *Traveling Tournament Problem* (TTP), em sua versão espelhada, conhecida como *Mirrored Traveling Tournament Problem* (mTTP). Foi apresentada uma proposta que faz uso conjunto de técnicas evolutivas, busca local e otimização baseada em restrições. É proposta uma representação compacta dos cromossomos (indivíduos da população), que passam por um algoritmo de expansão de código, responsável pela decodificação destes cromossomos em escalas de jogos, associado à aplicação de uma heurística gulosa para definição dos mandos de campo. Tal método se mostrou bastante eficaz, uma vez que toda solução obtida por ele é viável, fazendo com que a chance de se obter ótimos locais de boa qualidade aumente significativamente.

Pela análise dos resultados obtidos verificou-se a necessidade de se explorar outras estruturas de vizinhança, para fugir do molde que o Método do Polígono impõe as escalas geradas.

Enfim, existe uma variedade de assuntos em aberto, para serem pesquisados. Olhando pelo lado teórico eles estão mais em determinar quais estruturas de vizinhança estão conectadas e assim melhorar sua exploração. Já do lado prático está a necessidade de explorar novas estruturas de vizinhança e outras metaheurísticas.

Além das citadas, outras possíveis extensões deste trabalho podem ser realizadas pela implementação de operadores genéticos diferentes aos implementados neste trabalho, bem como a substituição da metaheurística *Simulated Annealing* por outro método de busca.

6. Referências

- [1] Anagnostopoulos, A.; Michel, L.; Van Hentenryck, P. & Vergados, Y. "A Simulated Annealing Approach to the Traveling Tournament Problem". In: *Proceedings of Cpaior'03*, 2003.

- [2] Biajoli, F. L. "Resolução do Problema de Programação de Jogos do Campeonato Brasileiro de Futebol". Monografia Final do Curso Ciência da Computação – DECOM, Universidade Federal de Ouro Preto, Ouro Preto – MG, 2003.
- [3] Biajoli, F. L.; Chaves, A. A.; Mine, O. M. & Souza, M. F. "Escala de Jogos de Torneios Esportivos: Uma Abordagem Via Simulated Annealing". In: XXXV Simpósio Brasileiro de Pesquisa Operacional 2003, Natal – RN. Anais XXXV SBPO, p. 1295–1306, 2003.
- [4] Biajoli, F. L.; Chaves, A. A.; Mine, O. M.; Souza, M. J. F.; Pontes, R. C.; Lucena, A. & Cabral, L. F. "Scheduling The Brazilian Soccer Championship: A Simulated Annealing Approach". In: Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat 2004, Pittsburgh, USA, 2004.
- [5] Concílio, R. & Zuben, F. J. "Uma Abordagem Evolutiva para Geração Automática de Turnos Completos em Torneios". In: *Revista Controle e Automação*. Vol.13, N.2, p. 105-122, 2002.
- [6] Costa, D. "An Evolutionary Tabu Search Algorithm and The NHL Scheduling Problem". In: *Infor*. vol. 33, n. 3, p. 161-178, 1995.
- [7] Dinitz, J. ; Lamken, E. & Wallis, W. D. "Scheduling a tournament". In C.J. Colbourn and J. Dinitz, editors, *Handbook of Combinatorial Designs*, p. 578–584. CRC Press, 1995.
- [8] Dowsland, K. A. "Simulated Annealing". In: C.R. Reeves, Editor, *Modern Heuristic Techniques for Combinatorial Problems, Advanced Topics in Computer Science Series*, Chapter 2, p. 20-69. Blackwell Scientific Publications, London, 1993.
- [9] Easton, K.; Nemhauser, G.; & Trick, M. "The Traveling Tournament Problem Description and Benchmarks". In: Seventh International Conference on the Principles and Practice of Constraint Programming (CP'99), p. 580–589, 2001.
- [10] Goldberg, D. E. "Genetic Algorithms in Search". *Optimization and Machine Learning*. Addison-Wesley: Berkeley, 223 p, 1989.
- [11] Hamiez, J. P. & Hao, J. K. "Solving The Sports League Scheduling Problem with Tabu Search". In: *Lecture Notes In Artificial Intelligence* 2148: p. 24-36, Springer, 2001.
- [12] Henz, M "Playing with Constraint Programming and Large Neighborhood Search for Traveling Tournaments". In: Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat 2004, Pittsburgh, USA, 2004.
- [13] Henz, M.; Muller, T.; & Thiel, S. "Global Constraints for Round Robin Tournament Scheduling". In: *European Journal of Operational Research*, vol. 153, p. 92–101, 2004.
- [14] Holland, J. H. "Adaptation in natural and artificial systems". Ann Arbor: University of Michigan Press, 211 p, 1975.
- [15] Kirkpatrick, S.; Gellat, D. C. & Vecchi, M. P. "Optimization by Simulated Annealing". In: *Science*, vol. 220, p. 671-680, 1983.
- [16] Miyashiro, R. & Matsui, T. "Round-Robin Tournaments with a Small Number of Breaks". In: *Mathematical Engineering Technical Reports*, Metr 2003-29, Department of Mathematical Informatics, Graduate School of Information Science and Technology, the University of Tokio, 2003.
- [17] Miyashiro, R.; Iwasaki, H. & Matsui, T. "Characterizing Feasible Pattern Sets with a Minimum Number of Breaks". In: Fourth International Conference on the Practice and Theory of Automated Timetabling, Patat 2002, p. 311-313, 2002.
- [18] Nemhauser, G. & Trick, M. "Scheduling a Major College Basketball Conference". In: *Operations Research* 46/1, p. 1–8, 1998.
- [19] Ribeiro, C.C. & Urrutia S. "Heuristics For The Mirrored Traveling Tournament problem". in: Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat 2004, Pittsburgh, USA, 2004.
- [20] Ribeiro, C.C. & Urrutia S. "Minimizing Travels by Maximizing Breaks in Round Robin Tournament". In: Schedules Latin-American Conference on Combinatorics, Graphs and Applications, Santiago, Chile, 2004.
- [21] Syswerda, G. "Uniform Crossover in Genetic Algorithms". In: International Conference on Genetic Algorithms (ICGA), vol. 3, 1989, Virginia. Fairfax: Morgan Kaufmann...George Mason University: Proceedings. p. 2–9, 1989.
- [22] Trick, M. A. "A Schedule-Then-Break Approach to Sports Timetabling". In: *Lecture Notes In Computer Science*. vol. 2079, p. 242-253, 2000.
- [23] Whitley, D.; Gordon, V. S. & Mathiask K. "Lamarckian Evolution, the Baldwin Effect and Function Optimization", in Davidor, Y., Schwefel, H.-P. & Männer, R. (eds.) Proceedings of the Third Conference on Parallel Problem Solving from Nature, Berlin: Springer, p. 6-15, October, 1994.
- [24] Zhang H.; Li D. & Shen, H. "A Sat Based Scheduler For Fair Tournament Schedules". In: The Seventh International Conference on Theory and Applications of Satisfiability Testing (Sat 2004), Vancouver, BC, Canada, p. 10-13, 2004.
- [25] Zhang, H. "Generating College Conference Basketball Schedules by a Sat Solver". In: Proceedings Of The Fifth International Symposium on the Theory and Applications of Satisfiability Testing, p. 281–291, Cincinnati, 2002.