

MAC 5758  
Introdução ao Escalonamento e  
Aplicações

---

**Escalonamento no Futebol Brasileiro**

Nome: Márcio Guedes Hasegawa

Nº USP: 4894760

# Índice

Introdução .....	2
Descrição do Problema .....	3
A Solução Proposta .....	4
As Vizinhanças .....	6
A Solução Inicial .....	9
A Perturbação .....	13
Outros Detalhes da Implementação .....	14
Resultados Obtidos .....	15
Referências .....	16

# Introdução

Uma das atividades de entretenimento que mais envolve dinheiro no mundo hoje são os torneios esportivos profissionais. Dada essa importância, é necessário que as equipes e as federações que organizam estes campeonatos façam o necessário para criar um produto cada vez melhor para apresentar aos seus consumidores.

Uma das formas de se melhorar a qualidade de um torneio é apresentar um agendamento de jogos que beneficie tanto os consumidores, quanto as equipes e os patrocinadores sem que a disputa se torne de alguma forma injusta.

Em 2001, Easton, Nemhauser e Trick propuseram o Traveling Tournament Problem, um problema de escalonamento de jogos esportivos cuja solução minimiza as distâncias viajadas pelas equipes ao longo do torneio. Com menos tempo gasto em viagens, as equipes teriam mais tempo para se preparar para os jogos seguintes e descansar, melhorando a qualidade técnica de forma geral.

Neste documento, será apresentada uma solução para um caso específico deste problema, associado ao modelo presente nas séries A e B do Campeonato Brasileiro de Futebol dos anos 2006, 2007, 2008 e 2009. Nestes torneios acontecem algumas viagens como por exemplo de Porto Alegre, Rio Grande do Sul, à Belém, Pará, que necessitam de vôos que percorram distâncias continentais.

## Descrição do Problema

O problema geral do Traveling Tournament Problem é um torneio que possui  $n$  equipes participantes, onde  $n$  é um número par, e cada uma delas deve jogar contra todos os outros  $n - 1$  adversários 2 vezes, sendo um confronto em casa e outro fora, em  $2 * (n - 1)$  rodadas compostas por  $n / 2$  jogos.

Para que o escalonamento das partidas não se torne muito benéfico ou prejudicial a alguma equipe, nenhuma equipe pode participar de mais de três jogos seguidos com mando de campo (jogando na localidade que definiu como sua casa no torneio) ou sem mando de campo (jogando na casa de seu oponente).

Sempre que uma equipe for participar de dois jogos consecutivos fora de casa, ela vai diretamente da localidade de um jogo para a localidade do outro sem retornar à sua sede para minimizar as viagens e uma equipe não pode participar de dois confrontos seguidos contra um mesmo adversário.

Todas as equipes iniciam o campeonato na localidade em que é sua sede, ou seja, se uma equipe tem o primeiro jogo fora de casa, esta viagem é computada e ao fim da competição, todos devem retornar às suas respectivas cidades.

A partir deste conjunto de regras, deve se encontrar o escalonamento de jogos que minimize a distância viajada pelas equipes ao longo de todo o campeonato.

Muitos autores propuseram diversas maneiras de se encontrar soluções para este problema geral, mas, em 2004, Ribeiro e Urrutia propuseram um caso específico do Traveling Tournament Problem que possuía uma maior correlação com os campeonatos de futebol latino-americanos.

A este problema, os autores deram o nome Mirrored Traveling Tournament Problem e o que este possui a mais como restrição é que a sequência dos  $n - 1$  primeiros jogos (turno) tem que ser a mesma dos jogos finais (retorno), apenas com o mando de campo invertido, portanto o adversário do jogo da rodada  $k$  de uma equipe será o mesmo da rodada  $k + (n - 1)$  para  $k = 1, 2, \dots, n - 1$ .

## A Solução Proposta

A solução proposta para o problema apresentado é uma hibridização de duas meta-heurísticas baseadas em busca local: *GRASP* e *ILS*.

A busca local é uma meta-heurística utilizada para resolver problemas de otimização computacionalmente muito custosos, assim como todas as meta-heurísticas, seu objetivo é encontrar uma solução próxima à solução ótima utilizando um tempo de execução com ordens de grandeza a menos. O procedimento executado baseia-se na aplicação de simples mudanças numa possível solução e a verificação se a nova solução criada, chamada de vizinha, é melhor que a anterior até que não se encontre soluções melhores.

Na meta-heurística *GRASP* (*Greedy Randomized Adaptive Search Procedure*), a idéia é se criar um número definido de soluções iniciais viáveis para o problema através de um processo baseado em aleatoriedade e a partir destas executar processos de busca local e selecionar a melhor dentre as soluções encontradas por cada uma das buscas.

Na meta-heurística *ILS* (*Iterated Local Search*), a idéia é se iniciar a partir de uma solução ótima local, executar sobre ela uma perturbação e, a partir da solução perturbada, realizar o processo da busca local para se encontrar uma nova solução ótima local, se a nova solução ótima local não for melhor que a anterior, ela é descartada. Este procedimento deve ser repetido até que algum critério de parada seja alcançado, como por exemplo, um número grande de repetições foi executado e não foi encontrada uma solução melhor ou algum critério associado a limite de tempo.

A solução híbrida, denominada *GRILS*, substitui a simples busca local utilizada em *GRASP* por um processo de *ILS*.

Pseudo-código:

```
GRILS(nIteracoes) {  
  para i = 1,..., nIteracoes {  
    S = solucaoInicial();  
    S', S = buscaLocal(S);  
    enquanto (criterioDeReinicializacao) {  
      S'' = perturbacao(S);  
      S'' = buscaLocal(S'');  
      S = criterioDeAceitacao(S, S'');  
      S* = atualizaMelhorSolucaoGlobal(S, S*);  
      S' = atualizaMelhorSolucaoLocal(S, S');  
    }  
  }  
}
```

O laço inicial presente no algoritmo executa  $n_{iteracoes}$  vezes o seguinte algoritmo: primeiro se contrói uma boa solução inicial, pois em um estudo, Elmohamed, Coddington e Fox chegaram a conclusão que algoritmos baseados em *simulated annealing* (recozimento simulado) têm performance fraca para criação de agendamentos, mas quando se apresenta um bom ponto de partida, há uma melhoria drástica na performance; a partir desta solução, se procura a melhor solução vizinha através de uma busca local e utiliza-se ela como entrada para o processo *LS* interno.

O laço interno começa se aplicando uma perturbação a solução atual e então se fazendo uma busca local na solução perturbada, então, o resultado passa pela avaliação do critério de aceitação, se aceito, é utilizado para atualizar as soluções local (desta iteração) e global (do algoritmo). O laço então é ciclado até que se cumpra o critério de reinicialização.

## As Vizinhanças

Para a execução da busca local, foram utilizadas três operações que definem vizinhanças: inversão de mando de campo, inversão de oponentes e inversão de rodadas.

- **Inversão de mando de campo**

As soluções participantes desta vizinhança são obtidas através da inversão do mando de campo de um jogo das primeiras  $n - 1$  rodadas e, por consequência, a inversão do jogo que envolve as mesmas duas equipes no retorno do campeonato.

Esta operação gera  $(n - 1) * n / 2$  soluções vizinhas a serem analisadas, nem todas elas são soluções viáveis ao problema.

1º turno:  $A \times B \rightarrow B \times A$

2º turno:  $B \times A \rightarrow A \times B$

- **Inversão de oponentes**

As soluções participantes desta vizinhança são obtidas pela inversão dos oponentes de todas as rodadas de duas equipes. Se um time for jogar em uma dada rodada contra um dos times escolhidos pela operação, ele passará a jogar contra o outro escolhido.

O agendamento dos jogos a serem realizados entre as equipes não tem como ser invertido, afinal uma equipe não pode jogar contra si própria.

Para que essa operação só gere vizinhanças viáveis, o mando dos jogos entre os times escolhidos é invertido.

Essa operação gera  $n * (n - 1) / 2$  soluções vizinhas a serem analisadas.

As tabelas 1 e 2 ilustram a inversão de oponentes entre os times A e B.

Rodada	A	B
1 e 6	F	E
2 e 7	D	C
3 e 8	B	A
4 e 9	E	D
5 e 10	C	F

→

Rodada	A	B
1 e 6	E	F
2 e 7	C	D
3 e 8	A	B
4 e 9	D	E
5 e 10	F	C

**Tabelas 1 e 2:** Inversão de oponentes

- **Inversão de rodadas**

As soluções participantes dessa vizinhança são obtidas através da inversão do agendamento dos jogos de duas rodadas. Todos os jogos que aconteceriam numa das rodadas escolhidas são agendados para ocorrer na outra rodada escolhida e vice-versa.

Após a operação realizada, são necessárias algumas trocas de mando de campo para que a solução encontrada seja viável, para isso é feita uma busca local utilizando como operação a inversão dos mandos de jogos.

Essa operação gera  $(n - 1) * (n - 2) / 2$  soluções vizinhas a serem analisadas.

As tabelas 3 e 4 ilustram a inversão de rodadas

	Times					
Rodada	A	B	C	D	E	F
1 e 6	F	E	D	C	B	A
2 e 7	D	C	B	A	F	E



	Times					
Rodada	A	B	C	D	E	F
1 e 6	D	C	B	A	F	E
2 e 7	F	E	D	C	B	A

**Tabelas 3 e 4:** Inversão de rodadas

## A Solução Inicial

Como citado anteriormente, para que seja obtida uma melhoria drástica no desempenho do algoritmo é necessária uma boa solução inicial, então será apresentado neste momento, o algoritmo utilizado para se alcançar este objetivo.

O algoritmo da solução inicial possui três passos bem definidos: o tabelamento de jogos de times fictícios, a associação dos times fictícios a times reais e a definição dos mandos de campo dos jogos.

- **Tabelamento de jogos de times fictícios**

Para criação do tabelamento de jogos, utilizaremos o "método do polígono", descrito por Dinitz, Lamken e Wallis em 1995.

O conceito é muito simples, utilizamos  $n$  equipes fictícias, retiramos uma e todas as outras  $n - 1$  passam a ser vértices de um polígono de  $n - 1$  lados. Um dos vértices é então associado ao participante removido e os outros são pareados de acordo com o número de lados do polígono a serem percorridos para se encontrá-los a partir do primeiro.

Guardada esta configuração, se rotaciona o polígono de forma que um dos vértices que compõe um dos lados do polígono junto com o que havia sido associado ao participante removido assumam a posição de associação ao excluído. Vão se guardando as associações até que o polígono retorne à sua configuração inicial.

A figura 1 ilustra o método do polígono para  $n = 6$ .

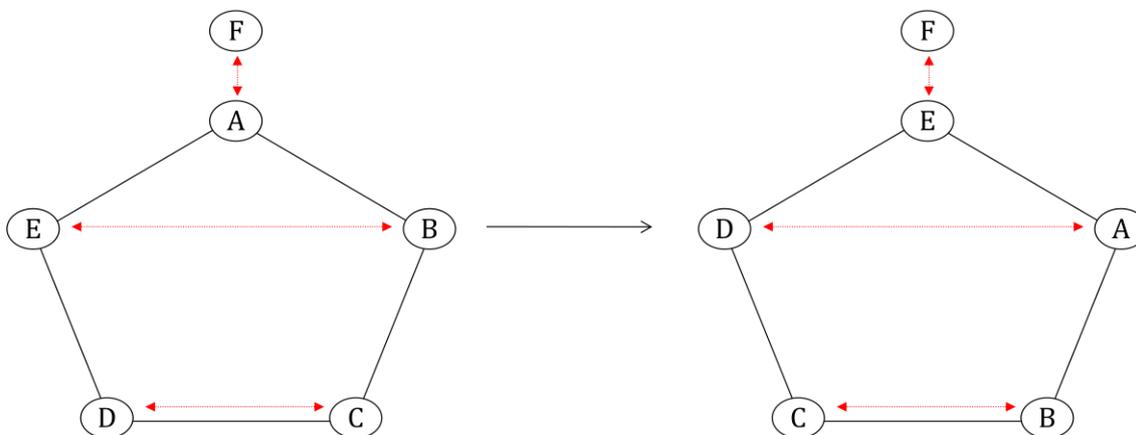


Figura 1: Ilustração do método do polígono

Com o agendamento dos  $n - 1$  primeiros jogos pronto, basta duplicá-lo para obter um escalonamento inicial para um Mirrored Traveling Tournament.

A tabela 5 mostra o escalonamento de jogos resultante.

	Times					
Rodada	A	B	C	D	E	F
1 e 6	F	E	D	C	B	A
2 e 7	D	C	B	A	F	E
3 e 8	B	A	E	F	C	D
4 e 9	E	D	F	B	A	C
5 e 10	C	F	A	E	D	B

Tabela 5: Escalonamento resultante do método do polígono

- **Associação de times reais aos fictícios**

Para se associar as equipes participantes do campeonato às equipes existentes no escalonamento inicial de jogos, será utilizada uma rápida heurística baseada nas distâncias entre as sedes dos times.

Inicialmente, criamos uma matriz cujos valores são associados à quantidade de jogos que uma equipe joga contra as mesmas duas ou mais equipes em sequência. As equipes que possuem sedes mais próximas, serão atribuídas às equipes fictícias que possuem valores altos nessa tabela.

Como, para que o algoritmo GRASP funcione corretamente, precisamos de uma solução inicial com certo nível de aleatoriedade, são feitas  $n / 8$  permutações entre as associações criadas entre as equipes reais e as fictícias.

A tabela 6 ilustra os valores retirados da tabela 5.

	Times					
Times	A	B	C	D	E	F
A	0	1	6	5	2	4
B	1	0	2	5	6	4
C	6	2	0	2	5	3
D	5	5	2	0	2	4
E	2	6	5	2	0	3
F	4	4	3	4	3	0

**Tabela 6:** Número de adversários seguidos entre um par de times

- **Definição de mando de campo**

Para a definição do mando de campo, utilizaremos um algoritmo de duas fases: primeiro criaremos um escalonamento de mandos que seja uma solução viável ao problema, depois utilizaremos busca local para se otimizar a solução encontrada.

Para a fase inicial, os mandos de campo das partidas da primeira rodada são definidos de forma aleatória. Nas rodadas 2 até  $n - 2$  utiliza-se o seguinte critério:

- Se o número de jogos seguidos em casa nas últimas rodadas de uma equipe é maior que o da outra, ela deve jogar fora de casa.
- Se o número de jogos seguidos fora de casa nas últimas rodadas de uma equipe é maior que o da outra, ela deve jogar em casa.
- Se ambas possuem as mesmas características de mando de campo nos últimos jogos, ele então é sorteado aleatoriamente.

A rodada  $n - 1$  sofre influência da rodada  $n$  que já tem os mandos de campo definidos pois serão os mandos de campo da primeira rodada invertidos, então atribui-se mandos de campo para esta rodada que tornem o escalonamento viável para o problema, se não houver tal combinação, todo o processo é repetido desde o início.

A fase final consiste em uma busca local que utiliza como operação para a definição das vizinhanças a inversão de mando de campo explicada anteriormente. A solução ótima encontrada nesta busca local é então utilizada como ponto de partida para a parte iterativa do algoritmo.

## A Perturbação

A perturbação é um dos passos de maior importância a serem executados no algoritmo *GRILS*. Através deste procedimento, faremos com que o algoritmo consiga escapar de soluções ótimas locais.

Soluções ótimas locais são aquelas em que todos os componentes de suas vizinhanças sejam avaliados como soluções piores que elas mesmas, mas que ainda assim não sejam a solução ótima global do problema.

O procedimento que iremos assumir como perturbação será o agendamento de um determinado jogo obrigatoriamente. Para isso, são sorteados aleatoriamente dois times e uma rodada do turno do torneio, então serão realizadas todas as alterações necessárias na tabela para que o jogo entre os times sorteados aconteça obrigatoriamente na rodada sorteada.

Esta é uma operação muito custosa pois precisam ser feitos vários ajustes e alguns deles não são triviais. Com a inserção do jogo sorteado, precisamos acertar a rodada para que as equipes não fiquem com o agendamento de dois jogos e na rodada em que aconteceria o jogo inicialmente, ele precisa ser retirado.

Mesmo após este acerto inicial, existe a necessidade de se acertar os mandos de campo para que a solução criada não seja inviável de acordo com as regras do problema, afinal, nenhuma equipe pode jogar mais que três jogos seguidos em casa ou fora.

Este subproblema é tão complexo que a solução proposta pelos autores é uma busca tabu (outra técnica de busca local, introduzida por Glover em 1989) para se encontrar uma solução satisfatória.

## Outros Detalhes da Implementação

- Critério de aceitação

Verifica se a solução viável testada é melhor ou pelo menos possui desempenho semelhante à melhor solução encontrada nesta iteração do GRASP. Soluções semelhantes são aceitas porque podem levar a soluções melhores nas próximas buscas locais.

- Critério de reinicialização

O critério de reinicialização é o mais simples possível, se forem executadas um número grande de perturbações e buscas locais sem que se encontre uma solução melhor para esta iteração, ela é encerrada. Portanto o algoritmo nunca é encerrado enquanto estiver encontrando soluções melhores.

## Resultados Obtidos

Os autores apresentaram alguns testes onde demonstram a qualidade da meta-heurística criada, comparando os resultados obtidos para ambientes conhecidos do Traveling Tournament Problem. Em alguns casos, este algoritmo alcançou soluções melhores que as obtidas por qualquer outro algoritmo na data da publicação.

Porém o foco do documento é a resolução do Mirrored Traveling Tournament Problem, apresentado no mesmo, que, portanto, não possuía nenhum resultado com o qual pudesse ser comparado este algoritmo.

Utilizando o programa para se testar com o agendamento oficial da série A do Campeonato Brasileiro de Futebol de 2003, chega-se a resultado animadores com uma redução de 52% da distância viajada pelas equipse, de 1.048.134 para 506.433 quilômetros.

## Referências

- K. Easton, G.L. Nemhauser, & M.A. Trick. The traveling tournament problem: Description and benchmarks. In T. Walsh, editor, Principles and Practice of Constraint Programming, volume 2239 of Lecture Notes in Computer Science, pages 580–584. Springer, 2001.
- 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.
- J. Dinitz, E. Lamken, & W.D.Wallis. Scheduling a tournament. In C.J. Colbourn and J. Dinitz, editors, Handbook of Combinatorial Designs, pages 578–584. CRC Press, 1995.
- Glover, F. Tabu Search — Part I, ORSA Journal on Computing, 1: 3, pages 190-206, 1989.
- Biajoli, F. L. & Lorena, L. A. N. Mirrored Traveling Tournament Problem: An Evolutionary Approach, 2006.