Bacharelado em Ciência da Computação Inteligência Artificial

Aluno(s): Guilherme Cardoso Silva
Renata Caroline Cunha

Instituto Federal de Minas Gerais-Campus Formiga

Junho de 2017

Conteúdo

Problema

Resolução do Problema de Programação de Jogos do Campeonato Brasileiro de Futebol

Um clássico problema em competições esportivas é a geração de escalonamentos satisfatórios para os confrontos entre os times.

Este problema é recorrente no campeonato brasileiro, onde todos os times se enfrentam em vários locais ao longo de um determinado período de tempo. Para isso, é necessário gerar uma tabela de confrontos que siga determinadas restrições do campeonato.

Devido ao elevado número de possibilidades a serem analisadas, a solução para esse problema é bastante complexa.

Por exemplo, o campeonato brasileiro, que possui 20 equipes participantes, há $2{,}9062*10^{130}$ combinações possíveis.

A geração da tabela implica tanto nos resultados dos jogos, quanto no rendimento financeiro da competição e nos gastos dos clubes.

Arquivos de Entrada

Os dados dos arquivos de entrada para o software de geração da tabela foram construídos a partir dos times participantes e das regras do campeonato brasileiro de 2017.

São eles: arquivos de dados (*.dat), arquivo de distâncias (*.dist) e arquivo de times (*.tim). A seguir será descrita cada uma destas estruturas de arquivo.

Arquivo de Dados (*.dat)

O arquivo *.dat contém o número de turnos, número de times, número de rodadas, caminho para o arquivo de distancias *.dist e o caminho para o arquivo de times *.tim

```
# Numero de turnos, Numero de Times, Numero de Rodadas
2,20,38
# Caminho para o arquivo de distancias *.dist
C:\dataset\distancias.dist
# Caminho para o arquivo de times *.tim
C:\dataset\times.tim
```

Arquivos de Distâncias (*.dist)

O arquivo de distâncias contém a distancia entre os times participantes do campeonato e a informação se são do mesmo estado.

```
#Indice do Time 1, Indice do Time 2, Distancia, Jogo Estadual
0,0,0,1
0,1,887,0
0,2,1211,0
0,3,1574,0
0,4,1646,0
0,5,1305,0
0,6,1536,0
```

Arquivos de Times (*.tim)

O arquivo de Times contém o nome de todos os times participantes do campeonato, seguindo a ordem dos índices que representarão estes times. Por exemplo 0 = Atletico-GO, 1 = Atletico-MG, 2 = Atletico-PR

```
#Times
Atletico-GO
Atletico-MG
Atletico-PR
```

Funcionamento do Algoritmo

O problema foi divido em 4 etapas, sendo elas:

- ***Etapa 1:*** Geração da Lista de Confrontos;
- ***Etapa 2:*** Geração de Solução Inicial;
- ***Etapa 3:*** Execução do Simulated Annealing em forma Adaptada;
- ***Etapa 4:*** Execução do Simulated Annealing em forma Básica.

Etapa 1: Geração da Lista de Confrontos

Nesta etapa é gerada uma lista de confrontos entre todos os times participantes do campeonato brasileiro, combinando-os em pares sem repetições de confrontos.

Para cada confronto gerado é sorteado o mando de campo para este jogo.

```
Atletico-GO x Atletico-MG
Atletico-GO x Atletico-PR       Atletico-MG x Atletico-PR
Atletico-GO x Avai              Atletico-MG x Avai              Atletico-PR x Avai
```

Etapa 2: Geração de Solução Inicial

Nesta etapa é gerada uma solução aleatória inicial preenchendo a matriz solução de acordo com o algoritmo a seguir.

```
Procedimento Constroi_Solucao_Inicial(L[.], Tabela[.][.], rMax)
1.  r ← 1;          {Representa a rodada a ser preenchida}
2.  j ← 1;          {Representa o jogo a ser preenchido}
3.  enquanto ( L <> ∅ ) faça
4.      jogo ← confronto sorteado da lista L de confrontos;
5.      Tabela[r][j] ← jogo;
6.      Atualizar lista L;
7.      if ( r < rMax) então  r ← r + 1;
8.      senão
9.              j ← j + 1;
10.             r ← 1;
11.     fim-se
12. fim-enquanto
13. Retorne  Tabela[.][.];
Fim Constroi_Solucao_Inicial;
```

Etapa 3: Simulated Annealing Adaptado

Nessa etapa o algoritmo Simulated será adaptado para realizar dois tipos de geração de vizinhos, sendo possível trocar o mando de campo de uma partida ou a troca de jogos entre rodadas. As regras analisadas nesta etapa são obrigatórias e por isto devem ser sempre atendidas. Para isto, enquanto a solução não for valida é feito um reaquecimento na temperatura.

Serão tratadas as seguintes regras:

- Um time não pode jogar mais de uma vez na mesma rodada;
- Nas duas primeiras rodadas, do turno e returno, que cada time participar, um jogo deve ser realizado em sua sede e outro fora. Por exemplo, se na primeira rodada do turno o confronto de um time for dentro de casa, então na segunda rodada o confronto deste time deve ser fora de casa;
- As duas últimas rodadas do turno, que cada time participar, devem ter a mesma configuração de seus dois primeiros confrontos. Exemplificando, se os dois primeiros confrontos de um time (turno) foram jogar fora de casa e depois em casa, respectivamente, então os dois últimos confrontos desse time (turno) devem ser, respectivamente, jogar fora de casa e depois em casa. A mesma regra vale para a fase de returno;
- Não pode haver jogos entre clubes do mesmo estado na última rodada.

```
Procedimento SA(f(.), N(.), α, SAmax, T0, s)
1.  s* ← s;                 {Melhor solução obtida até o momento}
2.  Iter ← 0;               {Número de iterações na temperatura T}
3.  T ← T0;                 {Temperatura corrente};
4.  enquanto ( T > temperatura de congelamento) faça
5.      enquanto (Iter < SAmax) faça
6.          Iter ← Iter + 1;
7.          Escolha aleatoriamente um dos movimentos definidos;
8.          Gere um vizinho qualquer s` ∈ N(s), a partir do movimento escolhido;
9.          Δ = f(s`) – f(s);
10.         se (Δ < 0 ) então
11.             s ← s`;
12.             se ( f(s`) < f(s*) ) então s* ← s`;
13.         senão
14.             Tome x ∈ [0, 1];
15.             se ( x < e^(-Δ/T)) então s ← s`;
16.         fim-se
17.     fim-enquanto
18.         T ← α * T;
19.         Iter ← 0;
20.         se ( T < TR ) então
21.             se ( inviabilidades essenciais não atendidas )  então
22.                 T ← 0.20 * (0.10 * T0);
23.                 SAmax ← 0.20 * SAmax;
24.             senão T ← TC – 1;  {Congela o sistema para terminar a execução}
25.             fim-se
26.         fim-se
25. fim-enquanto
26. s ← s*;
27. Retorne s;
Fim SA
```

Etapa 4: Simulated Annealing Básico

Nessa etapa o algoritmo Simulated irá realizar apenas troca entre os mandos de campos para satisfazer as regras restantes.

Entre estas regras, apenas a primeira é obrigatória, as outras devem ser atendidas sempre que possível.

As regras da Etapa 3 devem continuar sendo atendidas.

Serão tratadas as seguintes regras:

- O número de jogos realizados fora de casa deve ser igual ao número de jogos em casa dentro de cada turno;
- Evitar que um time jogue mais de duas vezes consecutivas em casa;
- Evitar que um time jogue mais de duas vezes consecutivas fora de casa;
- Minimizar distancia total percorrida pelos times;
- Diferença entre o time que se deslocou mais e o que se deslocou menos.

```
Procedimento SA(f(.), N(.), α, SAmax, T0, s)
1.  s* ← s;                 {Melhor solução obtida até o momento}
2.  Iter ← 0;               {Número de iterações na temperatura T}
3.  T ← T0;                 {Temperatura corrente};
4.  enquanto ( T > 0) faça
5.      enquanto (Iter < SAmax) faça
6.          Iter ← Iter + 1;
7.          Gere um vizinho qualquer s` ∈ N(s);
8.          Δ = f(s`) – f(s);
9.          se (Δ < 0 ) então
10.             s ← s`;
11.             se ( f(s`) < f(s*) ) então s* ← s`;
12.         senão
13.             Tome x ∈ [0, 1];
14.             se ( x < e^(-Δ/T)) então s ← s`;
15.         fim-se
16.     fim-enquanto
17.         T ← α * T;
18.         Iter ← 0;
19. fim-enquanto
20. s ← s*;
21. Retorne s;
Fim SA
```

Parâmetros de Execução

Parâmetros do Simulated Annealing Adaptado	
N° de Iterações	8000
Temperatura Inicial	10000000
Razão de Resfriamento	0.97

Parâmetros do Simulated Annealing Básico	
N° de Iterações	3000
Temperatura Inicial	100000
Razão de Resfriamento	0.95

Pesos Utilizados pelos Algoritmos	
Regra 1	100000000
Regra 2	1000000000
Regra 3	100000000
Regra 4	1000000000
Regra 5	2000000000
Regra 6	10000000
Regra 7	10000000
Regra 8	1
Regra 9	100

Resultados

Foram realizados 10 testes para os arquivos de entrada do ano de 2017.

Em 70% dos testes realizados obteve-se uma solução que atendeu a todos os requisitos.

Em 30% dos testes nem todas as regras serem atendidas, neste caso as regras 6 e 7, porém estas não são obrigatórias e devem ser atendidas sempre que possível.

Resultados

Regra	Execução 1	Execução 2	Execução 3	Execução 4	Execução 5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	1
7	1	1	0	0	1
8	747410	746114	741189	754066	729206
9	44816	55613	43631	48396	45029
Função Objetiva	25229010	26307414	5104289	5593666	45232106
Tempo	6min 51s	7min 54s	6min 54s	9 min 23s	6min 47s

Resultados

Regra	Execução 6	Execução 7	Execução 8	Execução 9	Execução 10
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	738282	745493	742670	755953	746648
9	40903	44746	48857	55327	45028
Função Objetiva	4828582	5220093	5628370	6288653	5249448
Tempo	8min 55s	9min 8s	8min54s	9min 14s	5min 7s

Melhores Resultados Encontrados

Melhores Valores da Função Objetivo				
Execução 6	Execução 3	Execução 7	Execução 10	Execução 4
4828582	5104289	5220093	5249448	5593666

Desempenho do Algoritmo	
Melhor valor da Função de Avaliação	4828582
Pior valor da Função de Avaliação	5593666
Media dos valores da Função de Avaliação	5199215,6

Resultado Tabela 2017

Regra	Tabela 2017
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	770465
9	54766
Função Objetiva	6247065

Analisando os resultados da tabela do campeonato brasileiro de 2017, pode-se observar que os resultados obtidos foram mais viáveis de acordo com as regras e pesos definidos nesta simulação.

Não se pode afirmar que os resultados obtidos são melhores que a tabela utilizada neste ano por falta de informações sobre os padrões analisados.

A melhor solução foi encontrada execução 6, sendo ela utilizada para comparação com o resultado da tabela de 2017.

Relação de Melhora da Melhor solução Comparado com a Tabela de 2017	
Regra 8	4,17%
Regra 9	25,31%
Função Objetiva	22,70%

Referências Bibliográficas

- BIAJOLI, F. L. Resolução do Problema de Programação de Jogos do Campeonato Brasileiro de Futebol. 2003. **Monografia Projeto Orientado (Monografia apresentada para obtenção do título em Bacharel em Ciência da Computação)–Universidade Federal de Ouro Preto. Ouro Preto**.