# 2 Programação de Tabelas e o Problema do Torneio com Viagens

As ligas esportivas profissionais representam atividades econômicas de importância no mundo todo. As equipes e as ligas investem em atletas e estrutura esperando obter um retorno que às vezes não é alcançado, eventualmente por causa da má organização dos jogos das competições.

O problema de programação de tabelas consiste em determinar quando e onde os jogos de um determinado campeonato serão realizados. Esta tarefa é apenas uma parte do planejamento de uma competição esportiva, já que a programação da tabela não determina nem o formato nem as regras da competição.

A programação de tabelas é uma tarefa difícil com múltiplas restrições e vários objetivos (envolvendo, entre outros, aspectos logísticos, de organização, econômicos e de equilíbrio entre as equipes participantes), assim como diferentes decisores (como autoridades das federações, administradores das equipes e executivos da televisão).

Este problema pode ser visto como um problema de seqüenciamento de tarefas, no qual estas são representadas pelos jogos e os recursos pelas equipes que participam do torneio. A única restrição comum a todos os problemas de programação de tabelas é que uma equipe não pode jogar mais de um jogo simultaneamente, isto é, o mesmo recurso não pode ser alocado para atender duas tarefas ao mesmo tempo.

Vários autores em diferentes contextos (ver por exemplo [3, 9, 44, 55, 57, 58, 66]) têm tratado o problema da programação de tabelas para diferentes ligas e esportes como futebol, basquete, hóquei, beisebol, râguebi, críquete e futebol americano, usando diferentes técnicas como programação inteira, programação por restrições, busca tabu, algoritmos genéticos e *simulated annealing*. Em [22] pode-se encontrar um *survey* deste tipo de trabalhos.

Em um torneio *round robin* simples (resp. duplo), todas as equipes têm que se enfrentar uma única vez (resp. duas vezes). Numerosos torneios são organizados desta forma e existem várias questões teóricas a resolver com respeito à estrutura deste tipo de torneio.

Quando os jogos de um torneio são sempre disputados na sede de uma das duas equipes que se enfrentam: uma das equipes joga em casa e a outra fora de casa. A equipe que joga em casa tem o *mando de campo* e é chamada de *mandante*, enquanto a outra equipe é chamada de *visitante*. Quando uma equipe têm dois jogos consecutivos em casa ou dois jogos consecutivos fora de casa, diz-se que essa equipe tem uma *quebra*. Problemas de minimização de quebras com diversas restrições têm sido estudados por vários autores. De Werra [12, 13, 14, 15, 16] mostrou como programar tabelas com número mínimo de quebras e provou outras propriedades dos problemas de programação de tabelas usando teoria de grafos. Elf et al. [23] consideraram o problema de minimizar quebras quando a seqüência dos jogos já está definida, faltando apenas determinar a sede de cada um desses jogos. Miyashiro et al. [42] caracterizaram completamente as tabelas com mínima quantidade de quebras para torneios de tipo *single round robin* com até 26 equipes.

A dificuldade dos problemas de programação de tabelas cresce quando as sedes das diferentes equipes estão geograficamente distribuídas em uma grande região. Considere-se, por exemplo, o campeonato brasileiro de futebol, do qual participam, entre outros, equipes de Porto Alegre e de Belém. Uma simples viagem entre essas cidades demora quase um dia inteiro, devido à ausência de vôos diretos. A distância viajada pelas equipes torna-se uma variável importante de ser minimizada, para diminuir as despesas em viagens e dar mais tempo de descanso aos jogadores.

Problemas de minimização de distância têm sido estudados por vários autores. Bean e Birge [4] trataram o problema de programação da tabela da *National Basketball Association* (NBA) dos Estados Unidos, no qual as restrições mais limitantes estavam relacionadas aos dias de descanso e à disponibilidade de estádios. Costa [9] considerou o problema de programação da tabela da *National Hockey League* (NHL) dos Estados Unidos, no qual um dos objetivos é minimizar a distância viajada pelas equipes durante o torneio. Há um interesse crescente nesta área, desde a formulação do Problema do Torneio com Viagens [2, 51].

O Problema do Torneio com Viagens apresenta algumas características dos problemas de programação de tabelas. Ele combina critérios de viabilidade com um problema de otimização difícil em termos computacionais, mas sua complexidade teórica ainda não foi estabelecida. O objetivo consiste em minimizar a distância percorrida pelas equipes durante o torneio, sujeito a que nenhuma equipe jogue mais de três jogos consecutivos em casa, nem mais de três jogos consecutivos fora. Como a distância total viajada durante o torneio é uma variável importante, resolver um Problema do Torneio com Viagens pode ser um bom ponto de partida para determinar-se uma solução de um problema real de programação de tabelas.

O trabalho presenteado em [22] é atualmente o *survey* mais completo e atual dos trabalhos produzidos na área nos últimos 35 anos, listando 45 publicações como referências.

## 2.1 Formalização do problema

Considera-se uma competição esportiva da qual participam $n \geq 2$ equipes. Essas equipes jogam um determinado número $m$ de jogos. Denota-se um jogo entre a equipe $i$ e a equipe $j$ pelo par $\{i, j\}$. Na sua versão mais simples, o problema de programação de tabelas consiste em dividir os $m$ jogos em $r$ rodadas, sujeito a que nenhuma equipe jogue mais de uma vez em cada rodada. Cada rodada tem no máximo $\lfloor n/2 \rfloor$ jogos. Uma determinada tabela é *compacta* se tem a mínima quantidade de rodadas possível para distribuir os $m$ jogos.

Em um torneio *round robin* simples (SRR, do inglês *single round robin*) cada equipe enfrenta as demais exatamente uma vez. Se $n$ é par, os $n(n-1)/2$ jogos de um torneio SRR compacto são divididos em $n-1$ rodadas com $n/2$ jogos cada uma. Se $n$ é ímpar, os jogos são divididos em $n$ rodadas com $(n-1)/2$ jogos cada uma.

Em um torneio *round robin* duplo (DRR, do inglês *double round robin*) cada equipe enfrenta as demais exatamente duas vezes. Se $n$ é par, os $n(n-1)$ jogos de um torneio DRR compacto são divididos em $2(n-1)$ rodadas com $n/2$ jogos cada uma. Se $n$ é ímpar, os jogos são divididos em $2n$ rodadas com $(n-1)/2$ jogos cada uma.

Quando o local de cada jogo é considerado, os problemas de programação de tabelas ganham mais uma dimensão. Além de determinar em que rodada cada jogo acontece, também deve-se determinar onde (campo, quadra, sede ou estádio) cada jogo será realizado. Neste caso, é usual ter-se uma restrição que não permite mais de uma certa quantidade de jogos por rodada em cada local.

Em campeonatos *inter-sedes*, cada equipe tem uma sede própria e cada jogo acontece na sede de uma das duas equipes que se enfrentam. A equipe que joga no seu próprio estádio tem um jogo em casa e a que joga no estádio do oponente tem um jogo fora de casa (ou, simplesmente, fora). Neste caso, a escolha do local onde o jogo será realizado se reduz a duas opções. Em cada rodada, cada estádio pode ser usado no máximo para um jogo. Quando o torneio é do tipo SRR ou DRR e compacto, a cada rodada a metade dos estádios sedia algum jogo e a outra metade nenhum. Nos torneios DRR inter-sedes, um dos dois jogos entre duas determinadas equipes acontece na sede da primeira e o outro na sede da segunda.

## 2.2 Fatoração de grafos

Um *grafo* $G = (V, E)$ é um objeto matemático composto por um conjunto de *nós* $V$ e um conjunto de *arestas* $E$. Cada aresta conecta um par de nós. Dois nós unidos por uma aresta são *adjacentes* e são *incidentes* à aresta que os une. Conceitos de teória de grafos não apresentados nesta tese podem ser encontrados e.g. em [5]. Um *subgrafo* de $G$ é um grafo $G' = (V', E')$ tal que $V' \subseteq V$ e $E' \subseteq E$. O *grau* $d_G(v)$ de um nó $v \in V$ é igual à quantidade de arestas de $E$ que tem $v$ como extremidade. Se as arestas do grafo têm uma orientação (chama-se nesta tese de *arcos* às arestas orientadas), o grafo é *orientado*. Neste caso, o par $(u, v)$ representa o arco que vai do nó $u$ ao nó $v$. Denota-se, nesta tese, um grafo orientado através de uma letra com uma seta acima e o conjunto de arcos do grafo por meio da letra $A$ para reforçar que as arestas são arcos, por exemplo: $\vec{G} = (V, A)$. Um grafo não orientado é *completo* se para todo par de nós distintos existe uma aresta. Um grafo orientado é completo se para todo par de nós distintos existem dois arcos, um do primeiro para o segundo e outro do segundo para o primeiro.

Dado um grafo $G = (V, E)$, um *fator* de $G$ é um subgrafo $G' = (V, E')$ com $E' \subseteq E$. Uma *fatoração* de $G$ é um conjunto de fatores $F = \{f_1, \ldots, f_p\}$, onde $f_i = (V, E_i)$ para $i = 1 \ldots p$, $E_i \cap E_j = \emptyset \ \forall\ 1 \leq i < j \leq p$ e $\bigcup_{i=1}^{p} E_i = E$. Um fator $f$ é um $[a, b]$-*fator* se $a \leq d_f(v) \leq b \ \forall\ v \in V$. Em uma $[a, b]$-*fatoração* de $G$, todos os fatores são $[a, b]$-fatores. Uma $[a, a]$-fatoração de $G$ é também chamada de uma *a-fatoração*. Quando uma certa ordem é atribuída aos fatores de uma fatoração, ela é chamada de *ordenada*. Em fatorações de grafos não orientados, pode-se atribuir uma orientação a cada aresta de cada fator. Neste caso, diz-se que a fatoração é *orientada*. Não deve-se confundir fatorações orientadas de grafos não orientados com fatorações de grafos orientados. Neste último caso, os arcos que são distribuídos em fatores já tem uma orientação única definida. Uma 1-fatoração orientada do grafo não orientado e completo de seis nós é apresentado na Figura 2.1.

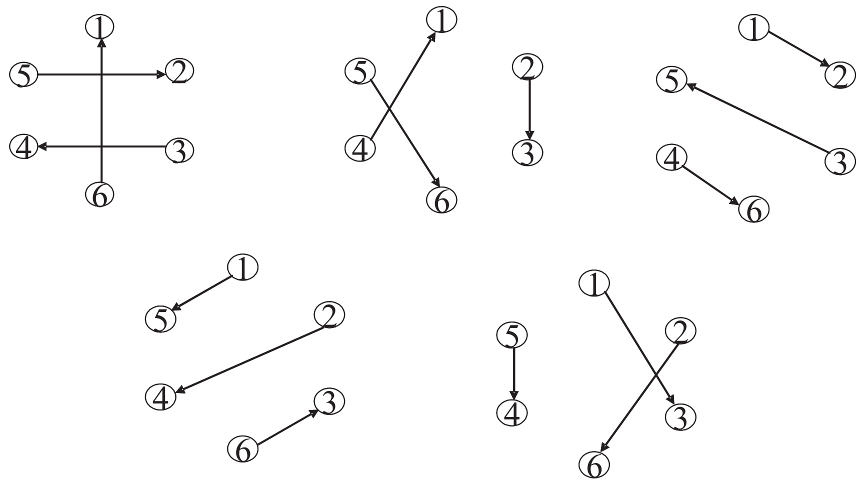

Figura 2.1: Uma 1-fatoração orientada de $K_6$



Pode-se associar um grafo a um torneio. Cada nó do grafo representa uma equipe participante. Cada aresta representa um jogo a ser disputado entre as equipes representadas pelos nós que lhe são incidentes. Cada $[0, 1]$-fator deste grafo representa uma possível rodada do torneio. O fato do grau de cada nó de um $[0, 1]$-fator ser menor ou igual a um garante que nenhuma equipe joga mais de um jogo por rodada. Em conseqüência, toda $[0, 1]$-fatoração desse grafo com $r$ $[0, 1]$-fatores, representa uma possível tabela para o torneio (neste caso, $r$ é o número de rodadas do torneio).

O grafo associado a um torneio SRR é completo. Se o torneio SRR é compacto e $n$ é par, então há $r = n - 1$ rodadas, todas as equipes jogam em todas as rodadas e toda tabela corresponde a uma 1-fatoração ordenada do grafo completo. 1-fatorações de grafos completos têm sido muito estudadas. Um bom *survey* dos trabalhos publicados no tema até 1985 é apresentado em [41]. Um *survey* mais recente pode ser encontrado em [63]. Se o torneio é inter-sedes, a tabela pode ser associada a uma 1-fatoração ordenada e orientada, de tal forma que o nó que é apontado por um arco represente a equipe que joga em casa no jogo correspondente.

O grafo associado a um torneio DRR inter-sedes é orientado e completo. Se o torneio DRR é compacto e $n$ é par, cada 1-fatoração ordenada deste grafo representa uma possível tabela do torneio.

Nota-se a diferença entre 1-fatorações ordenadas e orientadas de grafos completos não orientados (que representam torneios SRR compactos) e 1-fatorações ordenadas de grafos completos orientados (que representam torneios DRR compactos). No primeiro caso, trata-se de 1-fatorações com

$n-1$ 1-fatores, onde cada um desses 1-fatores contém $n/2$ arcos, cada um deles associado a uma das $n \cdot (n-1)/2$ arestas do grafo completo à qual é atribuída uma orientação. No segundo caso, trata-se de 1-fatorações com $2 \cdot (n-1)$ 1-fatores, onde cada um desses 1-fatores contém $n/2$ arcos, correspondendo cada um deles a um dos $n \cdot (n-1)$ arcos do grafo orientado completo.

## 2.3
## O problema do torneio com viagens

O problema do torneio com viagens (TTP, do inglês *Traveling Tournament Problem*) é um problema de programação de tabelas para campeonatos inter-sedes que concentra certas características dos problemas de seqüenciamento de jogos em competições esportivas [20]. Este problema combina aspectos de viabilidade com um problema de otimização difícil. Seu objetivo consiste em minimizar a distância total viajada pelas equipes durante o torneio, sujeito a que nenhuma equipe jogue mais de três jogos consecutivos em casa nem mais de três jogos consecutivos fora. Como a distância viajada é uma variável importante para as equipes participantes, resolver um problema do torneio com viagens pode ser um bom ponto de partida para a solução de um problema real de programação de tabelas.

Assume-se que cada uma das $n$ equipes participantes têm um estádio próprio na sua cidade e que as distâncias entre os estádios são conhecidas. As equipes estão na sua sede no começo do torneio, para onde todas retornam quando o torneio acaba. Quando uma equipe joga dois jogos consecutivos fora de casa, ela viaja diretamente da sede da primeira oponente para a sede da segunda.

Dado um número par de equipes $n$ e uma matriz quadrada $D$ de dimensão $n$, indicando por $D_{ij}$ a distância da sede da equipe $i$ até a sede da equipe $j$, o TTP consiste em construir uma tabela para um campeonato DRR inter-sedes compacto sujeito a que:

- nenhuma equipe pode jogar mais de três jogos consecutivos em casa nem mais de três jogos consecutivos fora;
- repetições consecutivas de jogos não são admitidas: se a equipe $t$ joga com a equipe $p$ na rodada $r$, elas não podem se reencontrar na rodada $r+1$;
- a somatória das distâncias percorridas pelas equipes durante o campeonato é minimizada.

O TTP pode ser interpretado como um problema definido sobre o grafo orientado completo $\vec{K}_n = (V, A)$ onde $V$ é o conjunto de equipes e $|V| = n$.

Dados o número de equipes $n$, uma matriz quadrada $D$ de distâncias de dimensão $n$ e uma 1-fatoração ordenada $\vec{F} = (\vec{f}_1, \dots, \vec{f}_{2(n-1)})$ de $\vec{K}_n$ associada a um torneio DRR, onde $\vec{f}_i = (V, A_i)$ para todo $i = 1, \dots, 2(n-1)$, definem-se as funções abaixo.

A função

$$d_c(v) = \begin{cases} D_{vu}, & \text{se } \exists\ u \in V \text{ tal que } (v, u) \in A_1, \\ 0, & \text{caso contrário;} \end{cases}$$

calcula a distância viajada pela equipe associada ao nó $v$ para jogar na primeira rodada do torneio associado à fatoração $\vec{F}$. Se no primeiro fator há um arco que sai de $v$, então a equipe associada a $v$ tem um jogo fora na primeira rodada. Portanto, deve viajar de sua sede à da seu oponente.

A função

$$d_f(v) = \begin{cases} D_{uv}, & \text{se } \exists\ u \in V \text{ tal que } (v, u) \in A_{2(n-1)}, \\ 0, & \text{caso contrário;} \end{cases}$$

calcula a distância viajada pela equipe associada ao nó $v$ para voltar à sua sede depois de jogar o último jogo. Se no último fator há um arco que sai de $v$, então a equipe associada a $v$ tem um jogo fora de casa na última rodada. Portanto, deve voltar à sua própria sede após esse jogo.

A função

$$d_i(v, q) = \begin{cases} D_{vw}, & \text{se } \exists\ u \in V \text{ tal que } (u, v) \in A_q \text{ e} \\ & \quad \exists\ w \in V \text{ tal que } (v, w) \in A_{q+1}, \\ D_{uv}, & \text{se } \exists\ u \in V \text{ tal que } (v, u) \in A_q \text{ e} \\ & \quad \exists\ w \in V \text{ tal que } (w, v) \in A_{q+1}, \\ D_{uw}, & \text{se } \exists\ u \in V \text{ tal que } (v, u) \in A_q \text{ e} \\ & \quad \exists\ w \in V \text{ tal que } (v, w) \in A_{q+1}, \\ 0, & \text{se } \exists\ u \in V \text{ tal que } (u, v) \in A_q \text{ e} \\ & \quad \exists\ w \in V \text{ tal que } (w, v) \in A_{q+1}; \end{cases}$$

calcula a distância viajada pela equipe associada ao nó $v$ entre as rodadas $q$ e $q + 1$. No primeiro caso, a equipe associada ao nó $v$ joga em casa na rodada $q$ e fora na rodada $q+1$, tendo que viajar de sua própria sede à sede do segundo oponente. No segundo caso, a equipe associada ao nó $v$ joga fora de casa na rodada $q$ e em casa na rodada $q+1$, tendo que viajar da sede do primeiro oponente à sua própria sede. No terceiro caso, a equipe associada ao nó $v$ joga fora de casa na rodada $q$ e na rodada $q+1$, tendo que viajar da sede do primeiro oponente à sede do segundo. No último caso, a equipe joga os dois jogos em casa, logo não viaja.

Então, a função

$$d_t(v) = d_c(v) + \sum_{q=1}^{2(n-1)-1} d_i(v,q) + d_f(v)$$

fornece a distância viajada pela equipe associada ao nó $v$ durante o torneio. O primeiro termo considera a distância viajada para jogar na primeira rodada. O somatório considera as distâncias viajadas pela mesma equipe da segunda até a última rodada. O último termo considera a distância viajada pela equipe para voltar à sua sede após a última rodada.

O TTP consiste em obter uma 1-fatoração ordenada de $\vec{K}_n = (V, A)$ tal que

a) $1 \leq |\{u : u \in V, (u,v) \in A_i \cup A_{i+1} \cup A_{i+2} \cup A_{i+3}\}| \leq 3 \quad \forall\ v \in V, 1 \leq i \leq 2(n-1) - 3$,

b) $(u,v) \in A_i \Rightarrow (v,u) \notin A_{i+1} \quad \forall\ u,v \in V, 1 \leq i \leq 2(n-1) - 1$;

e que minimize $\sum_{v \in V} d_t(v)$.

A restrição (a) exprime que em qualquer seqüência de quatro fatores consecutivos todo nó é origem de um arco e todo nó é destino de um arco. Desta forma, ela não permite que equipe alguma jogue mais de três jogos consecutivos nem em casa, nem fora de casa. A restrição (b) não permite repetições consecutivas de jogos. A função a minimizar corresponde à soma $\sum_{v \in V} d_t(v)$ das distâncias viajadas pelas equipes durante o torneio.

## 2.4
## O problema do torneio com viagens espelhado

Uma restrição habitual em competições esportivas, em especial na América do Sul, é que a tabela seja espelhada. Neste caso, o torneio DRR (chamado de MDRR, do inglês *mirrored double round robin*) está dividido em duas partes. Nas primeiras $n-1$ rodadas, as equipes participam de um torneio SRR. Nas últimas $n-1$ rodadas as equipes voltam a se enfrentar na mesma ordem em que nas primeiras $n-1$ rodadas, mas com os mandos de campo invertidos. Neste caso, diz-se que as primeiras $n-1$ rodadas formam o primeiro turno, enquanto que as $n-1$ últimas rodadas formam o segundo turno.

A versão espelhada do TTP inclui a restrição de que a tabela tem que ser espelhada. O conjunto de soluções viáveis para o problema do torneio com viagens espelhado (MTTP, do inglês *Mirrored Traveling Tournament Problem*) é um subconjunto do conjunto de soluções viáveis para o TTP (também chamado de TTP não-espelhado, para o qual são viáveis tabelas espelhadas e não-espelhadas) correspondente.

O MTTP também pode ser formulado como um problema em grafos. Para isto, a última restrição da formulação do TTP não-espelhado deve ser substituída pela seguinte restrição:

$$(u, v) \in A_i \Leftrightarrow (v, u) \in A_{i+(n-1)} \quad \forall\, u, v \in V, 1 \leq i \leq n-1$$

Observe-se que em uma tabela espelhada não há repetições de jogos em rodadas consecutivas.

Outra forma de formular o MTTP como um problema de grafos deriva-se da observação de que em torneios espelhados os jogos do segundo turno ficam determinados pelos jogos do primeiro. Pode-se formular o MTTP como se fosse um torneio SRR, com a função objetivo considerando que o torneio SRR é jogado duas vezes: na primeira, com as atribuições de mando de campo originais; na segunda, com os mandos de campo invertidos.

Nesse caso, o MTTP consiste em obter uma 1-fatoração orientada e ordenada de $K_n = (V, E)$, o grafo completo não orientado de $n$ nós, sujeito a restrições equivalentes às colocadas para o TTP não-espelhado e minimizando uma função também equivalente à definida para a versão não espelhada do problema.



## 2.5
## Espaço de busca

Duas fatorações $F$ e $H$ do mesmo grafo G=(V,E) são *isomorfas* se existe uma função bijetiva $\alpha : V \rightarrow V$ tal que se $\alpha$ é aplicada a todos os nós de todos os 1-fatores de $F$, a 1-fatoração resultante é $H$.

A quantidade $N(n)$ de 1-fatorações não-isomorfas de $K_n$ ($n$ par), cresce muito rapidamente: $N(2) = N(4) = N(6) = 1$, $N(8) = 6$, $N(10) = 396$ e $N(12) = 526,915,620$. Estima-se que $N(14) \approx 1.132 \times 10^{18}$, $N(16) \approx 7.07 \times 10^{30}$ e que $N(18) \approx 2.374 \times 10^{47}$, cf. [17, 18, 28].

Se cada nó de cada uma dessas 1-fatorações não-isomorfas é associado a uma equipe e uma determinada ordem é dada a cada 1-fator, as tabelas obtidas são todas diferentes. Para quase todas essas 1-fatorações ordenadas não-isomorfas, qualquer permutação das equipes associadas a cada nó gera uma nova tabela diferente. Cada permutação na ordem dos 1-fatores também gera uma tabela diferente.

Há algumas 1-fatorações (chamadas de simétricas ou automorfas) que são isomorfas a elas mesmas (com $\alpha$ diferente da identidade), de forma que existem pelo menos duas permutações dos nós que geram a mesma 1-fatoração. Essas 1-fatorações podem gerar tabelas iguais com duas associações diferentes de nós a equipes. Sabe-se, por exemplo, que a quantidade de 1-fatorações distintas de $K_{12}$ é 252,282,619,805,368,320 [18] e não $N(12) \times 12! \approx 2.52393 \times 10^{17}$.

Então, sem considerar-se as sedes onde os jogos acontecem, há da ordem de $N(n) \cdot n!(n-1)!$ tabelas diferentes para um campeonato SRR ou MDRR com $n$ equipes. Até $n = 12$ o número exato de tabelas diferentes é conhecido [7]. Os valores $1, 6, 720, 31449600, 444733651353600$ e $11! \cdot 252282619805368320$ correspondem à quantidade de tabelas diferentes para campeonatos SRR com 2, 4, 6, 8, 10 e 12 equipes, respectivamente. Considerando-se as sedes, o número de tabelas possíveis para campeonatos SRR ou MDRR é da ordem de $N(n) \cdot n!(n-1)! \cdot 2^{n(n-1)/2}$, já que isto corresponde a dar uma orientação a cada aresta da 1-fatoração. Embora nem todas estas tabelas sejam viáveis para o MTTP, devido à restrição de que as equipes não podem jogar mais de três jogos consecutivos em casa nem mais de três jogos consecutivos fora, este número dá uma idéia do tamanho do espaço de busca do problema.