4
Minimização de Viagens em Torneios Round Robin

Considera-se um torneio *round robin* compacto do qual participam n equipes (onde n é um número par maior do que dois), com as mesmas hipóteses feitas para a formulação do TTP na Seção 2.3.

Dada uma tabela de um torneio, o *padrão casa-fora* (HAP, do inglês *home-away pattern*) de uma equipe é um vetor de $n-1$ posições (resp. $2(n-1)$) no caso de um torneio SRR (resp. DRR) preenchidas com letras C ou F. Um C (resp. F) na posição r significa que esta equipe tem um jogo em casa (resp. fora de casa) na rodada r ou, em outras palavras, sua condição de jogo é mandante (resp. visitante). Um *conjunto HAP* associado com um torneio é uma coleção de n HAPs, cada um associado a uma equipe diferente, como exemplificado na Figura 4.1.

Equipe 1:	F	C	F	F	F
Equipe 2:	C	F	C	C	C
Equipe 3:	F	C	C	C	F
Equipe 4:	C	C	F	C	C
Equipe 5:	C	F	F	F	C
Equipe 6:	F	F	C	F	F

Figura 4.1: Conjunto HAP para um torneio SRR com seis equipes.

Uma equipe tem uma *quebra* na rodada r se ela tem dois jogos consecutivos em casa (quebra em casa) ou dois jogos consecutivos fora (quebra fora) nas rodadas $r-1$ e r. Dada uma tabela T, o número total de quebras $Q(T)$ é definido como a soma do número de quebras de todas as equipes na tabela. Dado que todo par de equipes tem que se enfrentar em alguma rodada, não podem existir dois HAPs iguais em um conjunto HAP. Dado que em toda rodada $n/2$ equipes jogam em casa e as demais $n/2$ equipes jogam fora, em qualquer rodada de qualquer tabela o número de quebras em casa é igual ao número de quebras fora de casa. Em conseqüência, o número de quebras em casa e o número de quebras fora são ambos iguais a $Q(T)/2$.

Dada uma tabela T, a distância total viajada $D(T)$ é definida como a soma das distâncias viajadas por todas as equipes durante o torneio.

Neste capítulo, estabelece-se uma conexão entre os problemas de minimização de distância e de maximização do número de quebras. Definem-se novas instâncias para problemas de minimização de distância, onde a distância entre todo par de equipes é igual a um, e mostra-se que para essas instâncias a minimização da distância viajada é equivalente à maximização do número de quebras. Esta relação motiva o estudo do problema de programação de tabelas com número máximo de quebras. Este problema foi previamente considerado por Russel e Leung [55] em um contexto onde as equipes não podem jogar mais de dois jogos consecutivos em casa nem mais de dois jogos consecutivos fora. Os autores mostraram que, nesse contexto, $n \cdot (n/2-1)$ é um limite superior para o número de quebras e desenvolveram um método para programar tabelas com $n \cdot (n/2-1) - (n-4)$ quebras.

Na Seção 4.2 mostra-se como construir tabelas para torneios SRR com o número máximo de quebras. Torneios DRR simples são considerados na Seção 4.3. Limites superiores para o número de quebras de torneios DRR espelhados que satisfazem às restrições do problema do torneio com viagens são obtidos na Seção 4.4. Esses limites superiores são usados na Seção 4.5 para a resolução exata das novas instâncias do problema do torneio com viagens espelhado.

4.1 Conectando o número de quebras com a distância viajada

Define-se a classe de *instâncias constantes* para problemas de minimização de distância como aquelas nas quais a distância entre qualquer par de sedes é igual a um. A distância viajada $D(T)$ é igual ao número de viagens para toda tabela T. As instâncias constantes serão usadas a seguir para estabelecer-se uma conexão entre a maximização de quebras e os problemas de minimização de distância.

As instâncias constantes são relevantes se as equipes têm como objetivo minimizar o número de viagens, em vez da distância viajada. Isto faz sentido quando, por exemplo, as equipes viajam de avião: neste caso, os custos não diferem tanto e cada viagem pode consumir muito tempo. Em outras palavras, quando os custos fixos de uma viagem são mais importantes do que os custos variáveis associados à distância viajada.

Quando o TTP foi formulado, as instâncias circulares foram definidas para avaliar se este problema seria mais fácil em instâncias onde o Problema do Caixeiro Viajante fosse trivial. Entretanto, experimentos computacionais revelaram que, na prática, as instâncias circulares parecem ser tão difíceis quanto as instâncias realistas. Como nas instâncias constantes toda solução viável é ótima para o Problema do Caixeiro Viajante, justifica-se pesquisar se o TTP é mais fácil nestas instâncias.

Nota-se que $n/2$ equipes fazem uma viagem na primeira rodada para jogar um jogo fora de casa. Para jogar nas demais rodadas, cada equipe faz uma viagem, exceto aquelas que têm uma quebra em casa nessa rodada. Dado que $n/2$ equipes terminam o torneio com um jogo fora de casa, elas têm que retornar a suas respectivas sedes após o último jogo. Seja $R = n-1$ (resp. $R = 2(n-1)$) o número de rodadas de um torneio SRR (resp. DRR). Então, a distância total viajada associada a uma tabela T é

$$D(T) = n/2 + n(R-1) - Q(T)/2 + n/2 = nR - Q(T)/2.$$

Então,

$$D(T) = \begin{cases} n(n-1) - Q(T)/2, & \text{se } T \text{ é uma tabela para um torneio SRR;} \\ 2n(n-1) - Q(T)/2, & \text{se } T \text{ é uma tabela para um torneio DRR.} \end{cases}$$

Pode-se então concluir que maximizando-se o número de quebras também se minimiza a distância viajada pelas equipes em instâncias constantes. Qualquer limite superior para o número de quebras da tabela determina um limite inferior para a distância viajada. Esta é a motivação para o estudo do problema de maximização de quebras. No contexto deste problema, procuram-se tabelas com a maior quantidade de quebras possíveis.

4.2 Número máximo de quebras para torneios SRR

Considera-se um torneio SRR sem restrições. O número máximo de quebras que uma equipe pode ter é $n-2$, caso jogue todos os jogos em casa ou todos os jogos fora. Uma equipe nesta situação tem quebras em todas as rodadas desde a segunda até a última. Dado que não existem HAPs repetidos associados às equipes participantes em um torneio, só duas equipes podem ter $n-2$ quebras: uma jogando todos seus jogos em casa e a outra jogando todos seus jogos fora. Então, as $n-2$ equipes restantes podem ter no máximo $n-3$ quebras cada uma. Logo,

$$LS_{SRR} = 2(n-2) + (n-2)(n-3) = n^2 - 3n + 2$$

é um limite superior para o número de quebras de uma tabela de um torneio SRR.

Utiliza-se a seguir o método do polígono (ver Seção 3.1.1) para gerar uma tabela de um torneio SRR com exatamente LS_{SRR} quebras.

Considera-se as arestas da 1-fatoração gerada pelo método do polígono. Uma extensão deste método atribui uma orientação a cada uma dessas arestas. A aresta conectando os nós 1 e n é orientada de 1 para n em todas as rodadas. Para cada $\ell = 2, \ldots, n/2$, a aresta conectando os nós ℓ e $n+1-\ell$ é orientada do nó par (resp. impar) para o nó impar (resp. par) nas rodadas pares (resp. ímpares). A equipe apontada por um arco da 1-fatoração ordenada e orientada gerada por este método é considerada como a que joga em casa. A Figura 4.2 ilustra este procedimento para $n = 6$.

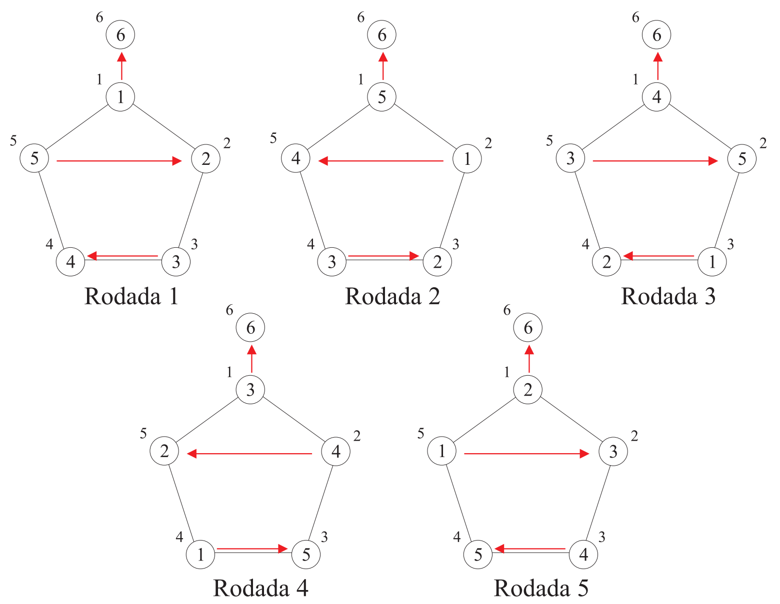

Figura 4.2: Método do polígono para maximizar quebras em torneios SRR para $n = 6$.

Todas as equipes de 1 até $n-1$ se movem sucessivamente entre nós pares e ímpares enquanto passam do nó 2 ao $n-1$. Em conseqüência, estão sempre ou em nós ímpares nas rodadas ímpares e em nós pares nas rodadas pares (sempre jogando fora), ou em nós pares nas rodadas ímpares e em nós ímpares nas rodadas pares (sempre jogando em casa). Então, as equipes de 1 a $n-1$ sempre têm quebras enquanto se movem do nó i ao nó $i+1$, para $2 \leq i \leq n-2$. Eles só podem trocar sua condição de jogo quando vão do nó $n-1$ ao nó 1 ou do nó 1 ao nó 2. Sempre que uma equipe vai sucessivamente do nó $n-1$ ao nó 1 e em seguida do nó 1 ao nó 2, ela fica em ambos nós, $n-1$ e 2, em rodadas ímpares ou em rodadas pares. Em conseqüência, joga em exatamente um desses nós fora e tem ou bem uma quebra na rodada na qual fica no nó 1 (se joga fora no nó $n-1$) ou no nó 2 (se joga fora no nó 2), dado que todas os equipes jogam fora na rodada na qual ficam no nó 1. Em conseqüência, todas as equipes de 1 até $n-1$ só podem ter uma rodada sem quebra e o número mínimo de quebras que podem ter é $n-3$.

Entretanto, dado que a equipe 1 é inicialmente colocada no nó 1, ela nunca passa do nó $n-1$ ao nó 1. Portanto, só pode mudar sua condição de jogo na segunda rodada quando deixa o nó 1. Dado que joga fora nas das primeiras rodadas, ela não muda sua condição de jogo durante todo o torneio e, em conseqüência, tem $n-2$ quebras. Finalmente, nota-se que a equipe n também tem $n-2$ quebras na tabela gerada com este método, dado que joga todos seus jogos em casa. Portanto, dado que só duas equipes têm $n-2$ quebras e nenhuma das outras pode ter menos do que $n-3$ quebras, a tabela tem exatamente $2(n-2)+(n-2)(n-3) = n^2-3n+2 = LS_{SRR}$ quebras.

Dado que as tabelas construídas por este método têm exatamente LS_{SRR} quebras e LS_{SRR} é um limite superior para o número de quebras, este método maximiza o número de quebras.

Consideram-se agora torneios SRR *equilibrados*, nos quais cada equipe joga $n/2-1$ ou $n/2$ jogos em casa. Isto é equivalente a dizer que cada equipe tem que jogar pelo menos $n/2-1$ jogos em casa e pelo menos $n/2-1$ jogos fora. Em conseqüência, nenhuma equipe pode ter $n-2$ quebras como no caso dos SRR irrestritos. Só quatro equipes podem ter $n-3$ quebras: uma jogando os primeiros $n/2$ jogos em casa e os seguintes $n/2-1$ jogos fora, a segunda jogando os primeiros $n/2-1$ jogos em casa e os seguintes $n/2$ jogos fora, a terceira jogando os primeiros $n/2$ jogos fora e os seguintes $n/2-1$ em casa, e a quarta jogando os primeiros $n/2-1$ jogos fora e os seguintes $n/2$ em casa. Portanto, as restantes $n-4$ equipes podem ter no máximo $n-4$ quebras. Então,

$$LS_{SRRE} = 4(n-3)+(n-4)(n-4) = n^2-4n+4 = (n-2)^2$$

é um limite superior para o número de quebras de uma tabela para um SRR equilibrado.

Para gerar uma tabela com exatamente LS_{SRRE} quebras, utiliza-se de novo o método do polígono. A aresta que une os nós 1 e n é orientada do nó 1 para o nó n nas primeiras $n/2-1$ rodadas e do nó n ao 1 nas últimas $n/2$ rodadas. Para cada $\ell = 2,\ldots,n/2$, a aresta que conecta os nós ℓ e $n+1-\ell$ é orientada do nó ℓ ao nó $n+1-\ell$ em todas as rodadas. A Figura 4.3 ilustra este método para $n = 6$ equipes.

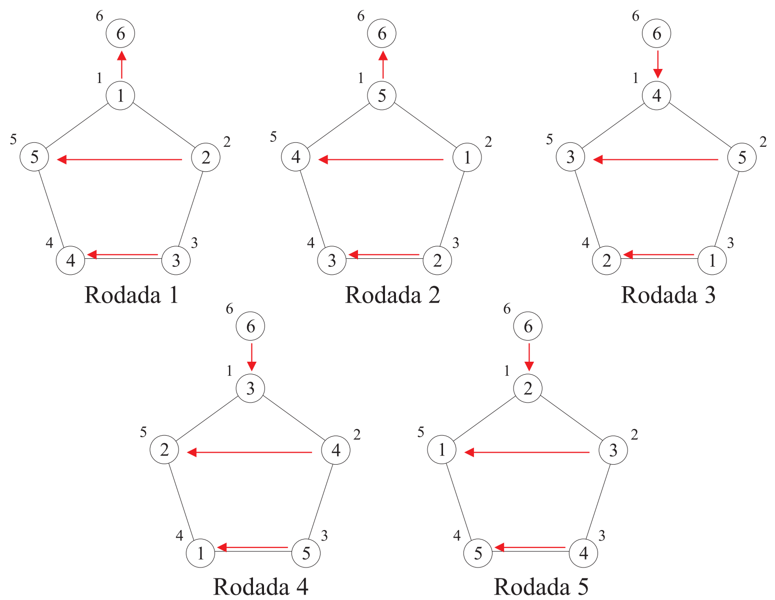

Figura 4.3: Método do polígono para maximizar quebras em torneios SRR equilibrados para $n = 6$.

Todas as equipes de 1 a $n - 1$ jogam fora de casa desde que entram no nó 2 e até que chegam ao nó $n/2$, tendo quebras quando se movem do nó i ao nó $i + 1$, para $2 \leq i \leq n/2 - 1$. Estas equipes jogam em casa desde que entram no nó $n/2 + 1$ e até que chegam ao nó $n - 1$, tendo quebras quando se movem do nó i ao nó $i + 1$, para $n/2 + 1 \leq i \leq n - 2$. Sempre que um equipe vai sucessivamente do nó $n - 1$ ao 1 e, em seguida, do nó 1 ao 2, tem-se uma quebra ou bem quando entra no nó 1 (se joga em casa no nó 1) ou bem quando entra no nó 2 (se joga fora no nó 1). Portanto, todas as equipes de 1 a $n - 1$ só podem ter duas rodadas sem quebra além da primeira: a primeira quando passam do nó $n/2$ ao $n/2 + 1$ e a segunda quando passam do nó $n - 1$ ao 1 ou do nó 1 ao 2. Em conseqüência, todas as equipes de 1 a $n - 1$ tem pelo menos $n - 4$ quebras.

Entretanto, a equipe 1 nunca passa do nó $n - 1$ ao 1 e joga fora na primeira e na segunda rodadas quando está nos nós 1 e 2, respectivamente. Então, ela muda sua condição de jogo só uma vez (quando passa do nó $n/2$ ao $n/2 + 1$) e tem $n - 3$ quebras. De forma similar, a equipe 2 nunca passa do nó 1 ao 2 e joga em casa quando está nos nós $n - 1$ e 1. Portanto, também muda sua condição de jogo só uma vez (quando passa do nó $n/2$ ao nó $n/2 + 1$) e também tem $n - 3$ quebras. A equipe $n/2 + 1$ nunca passa do nó $n/2$ ao nó $n/2 + 1$, então muda sua condição de jogo só uma vez e também tem $n - 3$ quebras. Finalmente, note-se que a equipe n também tem $n - 3$ quebras dado que só na rodada $n/2$ muda sua condição de jogo.

Portanto, dado que só quatro equipes têm $n - 3$ quebras e nenhuma das outras pode ter menos de $n-4$ quebras, o total de quebras na tabela é exatamente $4(n-3)+(n-4)(n-4)=n^2-4n+4=LS_{SRRE}$.

4.3
Número máximo de quebras para torneios DRR simples

Considera-se primeiro o problema no contexto de um torneio DRR sem restrição alguma. O número máximo de quebras que uma equipe poderia ter seria $2(n-1)-1=2n-3$, se jogasse todos os jogos fora ou todos os jogos em casa. Entretanto, em um torneio DRR todas as equipes jogam $n-1$ jogos em casa e $n-1$ jogos fora de casa. Para maximizar seu número de quebras, uma equipe deveria jogar os primeiros $n-1$ jogos em casa e os últimos $n-1$ jogos fora, ou vice-versa. A rodada n seria a única sem uma quebra para esta equipe. Só duas equipes podem ter $2n-4$ quebras. As $n-2$ equipes restantes podem ter no máximo $2n-5$ quebras. Então,

$$LS_{DRR}=2(2n-4)+(n-2)(2n-5)=2n^2-5n+2$$

é um limite superior para o número total de quebras de uma tabela de um torneio DRR.

Para gerar uma tabela com exatamente LS_{DRR} quebras, simplesmente duplica-se a tabela construída para o torneio SRR correspondente sem restrição alguma, com os mandos de campo invertidos nas últimas $n-1$ rodadas. As duas equipes que jogavam todos os seus jogos na mesma condição e tinham $n-2$ quebras cada uma no torneio SRR original, agora têm quebras em todas as rodadas com exceção da primeira e da rodada n, totalizando $2n-4$ quebras cada uma. As outras equipes têm quebras em todas as rodadas com exceção da primeira, a rodada onde não tinham quebra no torneio SRR original e a correspondente rodada na segunda parte do torneio DRR, totalizando 2n-5 quebras cada uma. Note-se que as tabelas construídas com este método são necessariamente espelhadas.

4.4
Limites superiores para o número de quebras para torneios MTTP

Considera-se agora o caso de torneios MDRR com as restrições do problema do torneio com viagens espelhado: nenhuma equipe pode jogar mais de três jogos consecutivos em casa nem mais de três jogos consecutivos fora.

Seja um *HAP compacto* (HAPC) uma seqüência de números inteiros positivos. Cada valor nesta seqüência representa o número de jogos consecutivos na mesma condição de jogo (mandante ou visitante) que uma equipe disputa na primeira metade de um torneio MDRR. Por exemplo, se uma equipe tem seu HAPC igual a (1,1,3) e joga seu primeiro jogo em casa, então joga o seu segundo jogo fora e os três últimos em casa na primeira metade de um torneio MDRR com seis equipes. Um HAPC deve satisfazer as seguintes condições para ser viável para um torneio com as restrições do MTTP:

– cada componente deve ser menor ou igual a três (dado que nenhuma equipe pode jogar mais de três jogos consecutivos em casa nem mais de três jogos consecutivos fora);
– a soma de todos os componentes deve ser igual a $n-1$ (dado que cada equipe joga exatamente $n-1$ vezes em cada metade do torneio DRR);
– se o numero de componentes é par, a soma do primeiro e do último componentes deve ser menor ou igual a três (dado que os jogos associados ao primeiro componente serão jogados na mesma condição na segunda metade do torneio que os jogos associados à última componente na primeira metade).

Cada HAPC representa dois possíveis HAPs, um com o primeiro componente associado a jogos em casa e o outro com o primeiro componente associado a jogos fora de casa. A Figura 4.4 mostra um HAPC e os seus dois HAPs correspondentes.

HAPC:	1	1	3		
HAP 1:	F	C	F	F	F
HAP 2:	C	F	C	C	C

Figura 4.4: Um HAPC e seus dois correspondentes HAPs.

Definem-se g_1, g_2 e g_3 como o número de componentes iguais a um, dois e três em um HAPC, respectivamente. Há uma quebra em cada seqüência de jogos associada a uma componente de valor dois. Há duas quebras em cada seqüência de jogos associada a uma componente de valor três. Portanto, e dado que o torneio é espelhado, o número de quebras de uma equipe com um dado HAPC é $2 \cdot (2g_3 + g_2)$ se $g1 + g_2 + g_3$ é ímpar e $2 \cdot (2g_3 + g_2) + 1$ caso contrário (dado que neste caso a equipe tem uma quebra na primeira rodada da segunda metade do torneio).

As instâncias deste problema serão separadas em três casos, de acordo com o número n de equipes no torneio:

- C_0: instâncias com $(n-1) \mod 3 = 0$
- C_1: instâncias com $(n-1) \mod 3 = 1$
- C_2: instâncias com $(n-1) \mod 3 = 2$

As instâncias em cada uma destas classes serão estudadas separadamente. Nota-se que os jogos devem ser agrupados em seqüências de jogos na mesma condição do maior tamanho possível para maximizar o número de quebras, que é diretamente proporcional a $2g_3 + g_2$. Além disso, sempre que possível, tentar-se-á ter um número par de componentes para maximizar o número de equipes com quebras na primeira rodada da segunda metade do torneio.

Limites superiores para as instâncias C_2:

Primeiramente, considera-se HAPCs com $g_3 = \lfloor (n-1)/3 \rfloor$, $g_2 = 1$ e $g_1 = 0$. Dado que $(n-1) \mod 3 = 2$, então $n \mod 3 = 0$ e $\lfloor (n-1)/3 \rfloor = n/3 - 1$. Dado que $n/3$ é par, $g_3 = \lfloor (n-1)/3 \rfloor = n/3 - 1$ é ímpar. Um HAPC com estes componentes tem $2 \cdot (2g_3 + g_2) + 1 = 2 \cdot (2\lfloor (n-1)/3 \rfloor + 1) + 1$ quebras. Contudo, este HAPC é inviável porque o número de componentes é par (g_3 é ímpar, g_2 é um e g_1 é zero) e a soma do primeiro e do último componente é maior do que três. Seja $Q = 2g_3 + g_2$, onde os valores de g_2 e g_3 se referem a este HAPC inviável.

Mostra-se agora que HAPCs com uma quebra a menos do que $2Q + 1$ são também inviáveis. Um HAPC com essa quantidade de quebras deverá ter necessariamente um número ímpar de componentes e $2g_3 + g_2 = Q$. Entretanto, isto não é possível porque para ter um número ímpar de componentes e manter $g_1 + g_2 + g_3 = n - 1$, pelo menos uma das seqüências deve ser dividida em seqüências menores.

Continua-se mostrando que HAPCs com duas quebras a menos que 2Q+1 são também inviáveis. Para obter um novo HAPC com exatamente duas quebras a menos, deve-se dividir seqüências de jogos de modo a obter um número par de componentes e $2g_3 + g_2 = Q - 1$. Isto não é possível porque pelo menos dois componentes adicionais devem ser criadas: ou bem uma seqüência de três jogos deve ser dividida em três seqüências de um jogo, ou bem mais de uma seqüência deve ser dividida. Em ambos casos, a soma $2g_3 + g_2$ seria menor que $Q - 1$.

Portanto, um HAPC nesta classe de instâncias pode ter no máximo $2(2\lfloor (n-1)/3 \rfloor + 1) + 1 - 3 = 4\lfloor (n-1)/3 \rfloor = 4(n/3 - 1)$ quebras. Em conseqüência, $n \cdot 4(n/3 - 1) = 4(n^2/3 - n) = LS_{C_2}$ é um limite superior para o número total de quebras em tabelas viáveis para instâncias da classe C_2.

<u>*Limites superiores para as instâncias C_1:*</u>

Mais uma vez, tenta-se agrupar, sempre que possível, jogos na mesma condição para maximizar $2g_3+g_2$ e, conseqüentemente, o número de quebras. Portanto, consideram-se HAPCs com $g_3 = \lfloor (n-1)/3 \rfloor$, $g_2 = 0$ e $g_1 = 1$ ou com $g_3 = \lfloor (n-1)/3 \rfloor - 1$, $g_2 = 2$ e $g_1 = 0$. Dado que $(n-1) \mod 3 = 1$, então $(n-2) \mod 3 = 0$, $\lfloor (n-1)/3 \rfloor = (n-2)/3$ e $\lfloor (n-1)/3 \rfloor$ é par. Em conseqüência, os dois tipos de HAPCs têm um número ímpar de componentes, são viáveis e têm $2 \cdot (2 \cdot \lfloor (n-1)/3 \rfloor) = 4\lfloor (n-1)/3 \rfloor = 4(n-2)/3$ quebras. Em conseqüência, $n \cdot 4(n-2)/3 = 4(n^2-2n)/3 = LS_{C_1}$ é um limite superior para o número de quebras de toda tabela viável para instâncias da classe C_1.

<u>*Limites superiores para as instâncias C_0:*</u>

Como antes, tenta-se agrupar os jogos de forma a maximizar $2g_3 + g_2$. Portanto, começa-se considerando HAPCs com $g_3 = (n-1)/3$, $g_2 = 0$ e $g_1 = 0$. Dado que $(n-1)/3$ é impar, este tipo de HAPC é viável e as equipes associadas têm $2(2(n-1)/3) = 4(n-1)/3$ quebras cada uma. Dado que só existe um HAPC com esses valores de g_1, g_2 e g_3 e sempre há dois possíveis HAPs para cada HAPC, só duas equipes podem ter este HAPC. Seja $Q = 2g_3 + g_2$, onde os valores de g_2 e g_3 se referem a este HAPC.

Para diminuir o número de quebras de uma unidade, a primeira possibilidade consiste em dividir uma seqüência de três jogos em uma seqüência de um jogo e outra seqüência de dois jogos. Desta maneira, o número de componentes torna-se par. Neste caso, considera-se HAPCs com $g_3 = (n-1)/3-1$, $g_2 = 1$ e $g_1 = 1$. Dado que o número de componentes é par, os dois componentes associados com as seqüências contendo um e dois jogos têm que ser necessariamente o primeiro e o último (em qualquer ordem) para que o HAPC seja viável e tenha $4(n-1)/3-1$ quebras. Dado que só há dois tipos de HAPCs viáveis satisfazendo estas condições (um começando com o componente igual a um e o outro começando com o componente igual a dois), só quatro equipes podem ter $4(n-1)/3-1$ quebras em esta situação. A segunda possibilidade consistiria em dividir duas seqüências de três jogos em três seqüências de dois jogos, com o número de componentes tornando-se par. Contudo, este tipo de HAPCs não seria viável porque a soma do primeiro e do último componente seria, necessariamente, maior do que três.

Nenhuma equipe pode ter $4(n-1)/3-2$ quebras. Essa equipe deveria ter um HAPC com a soma $2g_3 + g_2 = Q - 1$ e um número ímpar de componentes. Duas possibilidades existem: (a) uma seqüência de três jogos deveria ser dividida em uma seqüência de dois jogos e em uma seqüência de um jogo, ou (b) duas seqüências de três jogos deveriam ser divididas em três seqüências de dois jogos. Entretanto, em ambos casos o número de componentes torna-se par.

Similarmente, nenhuma equipe pode ter $4(n-1)/3-3$ quebras. Essa equipe deveria ter um HAPC com a soma $2g_3+g_2=Q-2$ e um número par de componentes. Como antes, duas possibilidades existem: (a) uma seqüência de três jogos deveria ser dividida em três seqüências de um jogo ou (b) duas seqüências de três jogos deveriam ser divididas em duas seqüências de dois jogos e duas seqüências de um jogo. Entretanto, em ambos casos o número de componentes se mantém ímpar e, portanto, o número de quebras seria $4(n-1)/3-4$.

Para $n=4$, duas equipes podem ter $4(n-1)/3$ quebras cada uma e as demais $4(n-1)/3-1$ quebras cada uma. Portanto, o número máximo de quebras está limitado por $2(4(n-1)/3)+2(4(n-1)/3-1)=14$. Para $n>4$, duas equipes podem ter $4(n-1)/3$ quebras cada uma, quatro equipes podem ter $4(n-1)/3-1$ quebras cada uma e as $n-6$ equipes restantes podem ter só $4(n-1)/3-4$ quebras cada uma. Neste caso, o número máximo de quebras está limitado por $2(4(n-1)/3)+4(4(n-1)/3-1)+(n-6)(4(n-1)/3-4) = 4(n^2-n)/3-4n+20$. Então,

$$LS_{C_0} = \begin{cases} 14, & \text{para } n=4; \\ 4(n^2-n)/3-4n+20, & \text{para } n>4 \end{cases}$$

é um limite superior para o número total de quebras de qualquer tabela viável para as instâncias da classe C_0.

Combinando-se os resultados acima para as três classes de instâncias, obtém-se o seguinte limite superior para o número de quebras em tabelas para torneios restritos pelas mesmas regras do MTTP:

$$LS_{MTTP} = \begin{cases} 14, & \text{se } n=4; \\ 4(n^2-n)/3-4n+20, & \text{se } (n-1) \mod 3 = 0 \text{ e } n \neq 4; \\ 4(n^2-2n)/3, & \text{se } (n-1) \mod 3 = 1; \\ 4(n^2/3-n), & \text{se } (n-1) \mod 3 = 2. \end{cases}$$

Dado que $D(T)=2n(n-1)-Q(T)/2$ nas instâncias constantes, o limite superior LS_{MTTP} para Q(T) pode ser usado no cálculo de limites inferiores para $D(T)$:

$$LI_{MTTP} = \begin{cases} 17, & \text{se } n = 4; \\ 4n^2/3 + 2n/3 - 10, & \text{se } (n-1) \mod 3 = 0 \text{ e } n \neq 4; \\ 4n^2/3 - 2n/3, & \text{se } (n-1) \mod 3 = 1; \\ 4n^2/3, & \text{se } (n-1) \mod 3 = 2. \end{cases}$$

4.5 Resolução do MTTP em instâncias constantes

Contrariamente aos casos mais simples de torneios SRR e DRR, um método construtivo geral para construir tabelas para torneios DRR espelhados com as restrições do MTTP e com exatamente LS_{MTTP} quebras não existe hoje em dia. Entretanto, soluções aproximadas podem ser obtidas explorando-se a conexão entre os problemas de maximização de quebras e de minimização de distância estabelecida na Seção 4.1.

Para resolver o MTTP nas instâncias constantes, aplica-se a heurística apresentada no Capítulo 3. Para cada instância constante com $n = 4, 6, \ldots, 20$, este algoritmo foi executado com um limite de 15 minutos de tempo de processamento em um computador Pentium IV com 2.0 GHz e 512 Mbytes de memória RAM.

A Tabela 4.1 apresenta os resultados numéricos. Para cada instância constante, definida pelo número n de equipes, mostra-se a distância viajada $D(T)$ associada à solução obtida pela heurística; o limite inferior para $D(T)$ derivado do limite superior LS_{MTTP}, explorando a conexão estabelecida na Seção 4.1; a diferença entre $D(T)$ e o limite inferior e o número de quebras $Q(T)$.

n	$D(T)$	LI	Dif.	$Q(T)$
4	17	17	-	14
6	48	48	-	24
8	80	80	-	64
10	130	130	-	100
12	192	192	-	144
14	256	252	4	216
16	342	342	-	276
18	434	432	2	356
20	526	520	6	468

Tabela 4.1: Melhores soluções para instâncias constantes do MTTP.

As instâncias constantes do problema do torneio com viagens espelhado com $n = 4, 6, 8, 10, 12$ e 16 equipes foram resolvidas até a otimalidade. A maior instância de qualquer versão do problema do torneio com viagens resolvida antes desta tese tinha apenas oito equipes [21]. Os limites superiores deduzidos na Seção 4.4 foram usados para provar a otimalidade das soluções obtidas pela heurística.

4.6 Conclusões

A criação de novas instâncias constantes para problemas de minimização de distância permitiu estabelecer uma relação entre este tipo de problema e o de maximização de quebras.

O problema de maximização de quebras foi resolvido de forma exata para torneios SRR, SRR equilibrados e DRR simples. Para torneios com as restrições do problema do torneio com viagens espelhado foram deduzidos limites superiores.

Esses limites superiores determinam, através da relação estabelecida, limites inferiores para as instâncias constantes do MTTP. Os limites inferiores foram usados para provar a otimalidade de várias das soluções encontradas pela heurística descrita no capítulo anterior.

As instâncias do MTTP resolvidas de forma exata neste capítulo são as maiores já resolvidas para qualquer variante do TTP.

Como conseqüência do trabalho desenvolvido neste capítulo foram publicados os artigos [61, 62].