

# An Improved Approximation Algorithm for the Traveling Tournament Problem with Maximum Trip Length Two\*

Mingyu Xiao<sup>1</sup> and Shaowei Kou<sup>2</sup>

1 University of Electronic Science and Technology of China, Chengdu, China  
myxiao@gmail.com

2 University of Electronic Science and Technology of China, Chengdu, China  
kou\_sw@163.com

---

## Abstract

The Traveling Tournament Problem is a complex combinatorial optimization problem in tournament timetabling, which asks a schedule of home/away games meeting specific feasibility requirements, while also minimizing the total distance traveled by all the  $n$  teams ( $n$  is even). Despite intensive algorithmic research on this problem over the last decade, most instances with more than 10 teams in well-known benchmarks are still unsolved. In this paper, we give a practical approximation algorithm for the problem with constraints such that at most two consecutive home games or away games are allowed. Our algorithm, that generates feasible schedules based on minimum perfect matchings in the underlying graph, not only improves the previous approximation ratio from  $(1 + 16/n)$  to about  $(1 + 4/n)$  but also has very good experimental performances. By applying our schedules on known benchmark sets, we can beat all previously-known results of instances with  $n$  being a multiple of 4 by 3% to 10%.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems

**Keywords and phrases** Sports scheduling, Traveling Tournament Problem, Approximation Algorithms, Timetabling Combinatorial Optimization

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2016.89

## 1 Introduction

In the field of tournament timetabling, the Traveling Tournament Problem is a well-known and practically difficult optimization problem inspired by Major League Baseball. This problem asks for a double round-robin schedule that minimizes the sum of distances traveled by all teams. Since the first introduction of this problem [6], several variants have been proposed, with a significant amount of research [13, 16]. Before introducing more background, we give the precise definition of the Traveling Tournament Problem.

### The Traveling Tournament Problem (TTP- $k$ ):

**Input:** An  $n \times n$  distance matrix  $D$  to indicate the distance between each pair of  $n$  teams, and an integer  $k$ ;

**Output:** A double round-robin tournament on the  $n$  teams such that the total distance traveled by all the teams is minimized, subject to the following three conditions:

---

\* This work is supported by National Natural Science Foundation of China, under the grant 61370071, and the Fundamental Research Funds for the Central Universities, under the grant ZYGX2015J057.



- *each-venue*: Each pair of teams plays twice, once in each other’s home venue.
- *at-most-k*: No team may have a home stand or a road trip lasting more than  $k$  games.
- *no-repeat*: A team cannot play against the same opponent in two consecutive games.

When calculating the total distance, we assume that each team begins the tournament at home and returns home after playing its last away game. Furthermore, whenever a team has a road trip consisting of multiple away games, the team does not return to its home city but rather proceeds directly to its next away venue. There are two commonly used assumptions: each team has a game scheduled on each time slot with no time slots off and then the number  $n$  of teams must be even. The distances in  $D$  satisfy symmetry and triangle inequality, i.e.,  $D_{i,j} = D_{j,i}$  and  $D_{i,j} \leq D_{i,h} + D_{h,j}$  for all  $i, j, h$ . We also require that  $D_{i,i} = 0$  for each  $i$ .

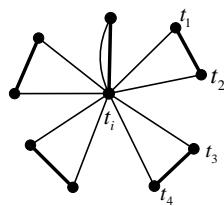
The integer  $k$  in the input defines the tradeoff between distance traveled and the length of the home stands and road trips. For the case that  $k = \infty$ , there is no constraint on the number of consecutive home stands or road trips and a team can be scheduled with its traveling length as short as that of the traveling salesman tour of the cities. The smaller  $k$ , the more often teams have to return to their home cities.

TTP- $k$  can be regarded as a variant of the well-known Traveling Salesman Problem. The NP-hardness of TTP- $k$  with  $k = \infty$  or  $k = 3$  has been proved [17, 2]. There is a large number of contributions on approximation algorithms [22, 12, 15, 21, 11] and heuristic algorithms [7, 14, 1, 5, 9].

There is an online set of benchmark data sets [19] with the list of best-known results for TTP-3. For most benchmark problems, instances are often completely solved or improved after weeks of computation on high-performance machines using parallel computing, see, e.g. [20]. Since the search space of TTP- $k$  is very large, no instance with more than 10 teams in [19] has been completely solved even on high-performance machines. Goerigk *et al.* [9] used a technique of packing  $P_3$  paths to find feasible solutions as initial inputs for some hybrid algorithms and then improved five benchmark instances of TTP-3. New techniques become more important to get further improvements.

In this paper, we focus on TTP-2. TTP-2 was first mentioned by Campbell and Chen [3], who scheduled a basketball conference of ten teams. In this problem, all away trips consist of either a single team or pairs of teams. It is reasonable that each team has at most two consecutive home stands or road trips in practice. In a schedule, we hope that home stands and road trips alternate as regularly as possible for each team. We can see that the perfect schedule with  $k = 1$  can not be achieved [4]. It is natural to consider TTP-2. However, compared to the case that  $k \geq 3$ , TTP-2 did not attract much attention. A significant contribution to approximation algorithms for TTP-2 is due to Thielen and Westphal [18]. They first gave an approximation algorithm with ratio  $3/2 + O(1/n)$  and then improved the ratio to  $1 + 16/n$  for the case that  $n \geq 12$  and  $n$  is a multiple of 4. Their algorithms also get the current best results on the benchmark instances listed on the website [19].

The main contribution of this paper is an improved approximation algorithm for TTP-2. Our algorithm generates a feasible solution to TTP-2 for  $n$  being a multiple of 4 with a traveling distance at most  $(1 + 2/(n - 2) + 2/n)$  times of the optimal distance, improving the previous approximation ratio of  $(1 + 16/n)$  by an addition of almost  $12/n$ . Our algorithm takes only 2.6 seconds on a standard laptop to compute all the instances with  $n$  being a multiple of 4 (half of all the benchmark instances) in the benchmark [19], and beat the previously best-known upper bounds by 3% to 10%.



■ **Figure 1** the itinerary graph of team  $t_i$ .

The remaining parts of the paper are organized as follows: We first introduce a simple lower bound of the problem, then introduce our methods of constructing schedule, and finally prove the approximation ratio and demonstrate the performance of the algorithm on benchmark instances.

## 2 A Lower Bound

Most lower bounds for the Traveling Tournament Problem are obtained by a relaxation technique called “independent”. It is to compute the minimum distance of a “feasible” traveling for each team independently and then sum all of them together to get a bound. Here “feasible” means that the traveling satisfies the three conditions in the definition of TTP- $k$ . This bound is known as the “independent lower bound”.

The independent lower bound for TTP-2 was firstly obtained by Campbell and Chen [3]. It can be computed by finding a minimum perfect matching in a complete undirected graph  $G$  on the set of teams with edge weight being the distance between the homes of two teams. By triangle inequality, we know that an optimal feasible traveling for a team contains at most one away trip of a single team and all other away trips of a pair of teams. According to the definition of the problem, the number  $n$  of teams is even. So each team contains exactly one away trip consisting of a single site (team) and all other away trips consisting of a pair of sites (teams). The itinerary graph of a team  $t_i$  is as shown in Figure 1.

Each team  $t_i$  must travel to or from each other team for at least once. See the light lines in Figure 1. The total length of all light lines is a constant. It is the total distance from a team  $t_i$  to all other teams, which is also denoted by  $D_i$ . We have that  $D_i = \sum_{j \neq i} D_{i,j}$ .

The dark lines in Figure 1 form a perfect matching of  $G$ . We use  $M$  to denote a minimum perfect matching (a perfect matching with minimum total edge weight) of  $G$  and use  $D_M$  to denote the total edge weight of  $M$ . We can observe that the traveling distance of team  $t_i$  is at least

$$LB_i = D_i + D_M,$$

which is called the *independent lower bound* for team  $t_i$ . We use  $D_G$  to denote the sum of the weights of all edges in  $G$ . A lower bound for the Traveling Tournament Problem, obtained by summing up the independent lower bound of each team, is given as follows.

$$LB = \sum_{i=1}^n LB_i = \sum_{i=1}^n (D_i + D_M) = 2D_G + nD_M. \quad (1)$$

If we can find a feasible tournament schedule such that all teams achieve the independent lower bound synchronously, then the Traveling Tournament Problem is solved optimally. However, it is impossible for all teams to reach the independent lower bound synchronously in any feasible tournament schedule [18]. It is also worthy to mention that for  $k \geq 3$  it is

even NP-hard to compute the independent lower bound for a team, since it will involve the problem of finding an optimal  $k$ -path packing in a graph.

### 3 Techniques for Construction

It is nontrivial to obtain a feasible tournament schedule for TTP-2 even without considering the traveling distance. “Expander construction” is an effective method used to construct feasible schedules for TTP-3 [10, 9]. We will modify this method for TTP-2 to construct an initial solution. After obtaining a feasible tournament schedule, we use some techniques based on minimum perfect matchings to arrange the order of teams and then we can obtain a solution with the traveling distance quite near to the independent lower bound.

To make the traveling distance small, we hope that an away trip of a team consists of a pair in a minimum perfect matching of  $G$ . This gives us an idea to consider the teams in the tournament as pairs corresponding to a minimum perfect matching. After scheduling the pairs, we “expand” by replacing each pair with two original teams to get the final schedule. This is the main idea of the initial construction.

The construction contains two steps. Step 1 is to create a single round-robin tournament  $U_m$  on  $m = n/2$  teams (each of which will represent a pair of original teams). Step 2 is to expand  $U_m$  to a double round-robin tournament  $Z_n$  on  $n$  teams. Note that the construction only works for  $m = n/2$  being even, i.e.,  $n \equiv 0 \pmod{4}$ . Next, we always assume that  $m$  is even.

**Step 1. Constructing a single round-robin tournament  $U_m$ :** The single round-robin tournament  $U_m$  on  $m$  teams is built by using a variation of the Modified Circle Method [8, 10]. We use  $\{u_1, u_2, \dots, u_{m-1}, x\}$  to denote the  $m$  teams. Each team plays with each of the other  $m - 1$  teams on  $m - 1$  time slots according to the following rule: for each  $1 \leq i \leq m - 1$ , team  $u_i$  plays with team  $u_j$  on time slot  $r$  such that

$$r - i \equiv j - 1 \pmod{m - 1},$$

where we interpret the case that a team  $u_i$  plays with itself as that  $u_i$  plays with team  $x$  on the time slot. This assignment can guarantee a feasible schedule, i.e., each of the  $m$  teams plays with another team on each of the  $m - 1$  time slots.

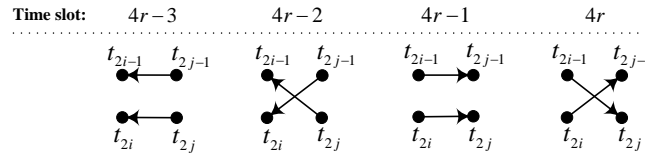
The construction is not finished yet. We still designate a home team and a road team for each game not involving team  $x$ : for each  $1 \leq i \leq m/2$ ,  $u_i$  plays only road games until it meets team  $x$ , before finishing the remaining games at home; for each  $m/2 + 1 \leq i \leq m - 1$ , we have the opposite scenario, where  $u_i$  plays only home games until it meets team  $x$ , before finishing the remaining games on the road. Please see Table 1 for an illustration of the single round-robin schedule with  $m = 8$ , where items in bold font indicate that the corresponding teams (on the left of the table) are home teams in this game.

**Step 2. Constructing a double round-robin tournament  $Z_n$ :** We have four substeps to construct a double round-robin tournament  $Z_n$  on  $n$  teams from the single round-robin tournament  $U_m$  on  $m = n/2$  teams. Recall that each team  $u_i$  in  $U_m$  is represented with a pair of original teams in the tournament. So we will replace  $u_i$  (where  $x$  is interpreted as  $u_m$ ) with two original teams  $\{t_{2i-1}, t_{2i}\}$  in this step. Thus, the set of the original  $n$  teams in  $Z_n$  is denoted by  $\{t_1, t_2, \dots, t_{n-1}, t_n\}$ .

In  $U_m$ , a game on the last time slot is called a *last game* and a game not on the last time slot is called a *normal game*. To construct  $Z_n$ , we distinguish four kinds of games in  $U_m$  according to the game being a last game or not and involving team  $x$  or not.

■ **Table 1** The single round-robin construction for  $m = 8$ .

	1	2	3	4	5	6	7
$u_1$	$\mathcal{D}$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$
$u_2$	$u_7$	$u_1$	$\mathcal{D}$	$u_3$	$u_4$	$u_5$	$u_6$
$u_3$	$u_6$	$u_7$	$u_1$	$u_2$	$\mathcal{D}$	$u_4$	$u_5$
$u_4$	$u_5$	$u_6$	$u_7$	$u_1$	$u_2$	$u_3$	$\mathcal{D}$
$u_5$	$u_4$	$\mathcal{D}$	$u_6$	$u_7$	$u_1$	$u_2$	$u_3$
$u_6$	$u_3$	$u_4$	$u_5$	$\mathcal{D}$	$u_7$	$u_1$	$u_2$
$u_7$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$\mathcal{D}$	$u_1$
$x$	$u_1$	$u_5$	$u_2$	$u_6$	$u_3$	$u_7$	$u_4$



■ **Figure 2** Expansion of Case 1.

■ **Table 2** Expanding a game of Case 1.

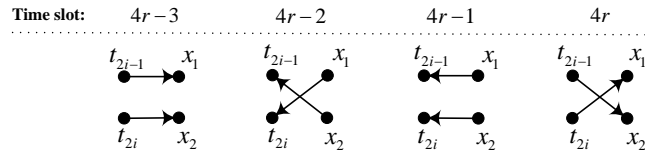
	$4r-3$	$4r-2$	$4r-1$	$4r$
$t_{2i-1}$	$t_{2j-1}$	$t_{2j}$	$t_{2j-1}$	$t_{2j}$
$t_{2i}$	$t_{2j}$	$t_{2j-1}$	$t_{2j}$	$t_{2j-1}$
$t_{2j-1}$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$
$t_{2j}$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$

**Case 1. Normal games not involving team  $x$ :** We consider a game in  $U_m$ , where a home team  $u_i$  plays against a road team  $u_j$  on time slot  $r$  ( $1 \leq i, j \leq m - 1$  and  $1 \leq r \leq m - 2$ ). We will expand this game to  $2 \times 4 = 8$  games on four consecutive time slots in  $Z_n$ . The corresponding four time slots are from  $4r - 3$  to  $4r$ . Recall that  $u_i$  will be replaced with  $\{t_{2i-1}, t_{2i}\}$  and  $u_j$  will be replaced with  $\{t_{2j-1}, t_{2j}\}$ . Figure 2 demonstrates how the four teams play on the four time slots, where an arc from  $a$  to  $b$  means a road team  $a$  playing against a home team  $b$ .

The eight games in Figure 2 determine 16 items in  $Z_n$ , which correspond to the eight games between four teams  $\{t_{2i-1}, t_{2i}, t_{2j-1}, t_{2j}\}$  on the four time slots from  $4r - 3$  to  $4r$ . The matching assignments in  $Z_n$  are presented in Table 2.

Note that in this scheduling, each of  $\{t_{2i-1}, t_{2i}\}$  has an away trip consisting of two teams in  $\{t_{2j-1}, t_{2j}\}$ , and also each of  $\{t_{2j-1}, t_{2j}\}$  has an away trip consisting of two teams in  $\{t_{2i-1}, t_{2i}\}$ . Furthermore, there is no conflict to assign the games in  $Z_n$  corresponding to all games of Case 1 in  $U_m$ , i.e., the three conditions in the definition of TTP- $k$  hold.

**Case 2. Normal games involving team  $x$ :** We consider a game in  $U_m$ , where a team  $u_i$  plays against the team  $x$  in time slot  $r$  ( $1 \leq i \leq m - 1$  and  $1 \leq r \leq m - 2$ ). For the purpose of presentation, we use  $x_1$  and  $x_2$  to denote the two teams in  $Z_n$  corresponding to  $x$ , i.e.,  $x_1 = t_{n-1}$  and  $x_2 = t_n$ . We also expand this game to  $2 \times 4 = 8$  games on four consecutive time slots in  $Z_n$ . However, the expansions are different according to the time slot  $r$  being odd or even.



■ **Figure 3** Expansion of Case 2 on an odd time slot.

■ **Table 3** Expanding a game of Case 2 on an odd time slot.

	$4r - 3$	$4r - 2$	$4r - 1$	$4r$
$t_{2i-1}$	$x_1$	$x_2$	$x_1$	$x_2$
$t_{2i}$	$x_2$	$x_1$	$x_2$	$x_1$
$x_1$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$
$x_2$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$

■ **Table 4** Expanding a game of Case 2 on an even time slot.

	$4r - 3$	$4r - 2$	$4r - 1$	$4r$
$t_{2i-1}$	$x_1$	$x_2$	$x_1$	$x_2$
$t_{2i}$	$x_2$	$x_1$	$x_2$	$x_1$
$x_1$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$
$x_2$	$t_{2i}$	$t_{2i-1}$	$t_{2i}$	$t_{2i-1}$

On an odd time slot  $r$ , a team  $u_i$  with  $1 \leq i \leq m/2$  plays against the team  $x$ . We assign the games among four teams  $\{t_{2i-1}, t_{2i}, x_1, x_2\}$  on time slots from  $4r - 3$  to  $4r$  according to Figure 3.

The corresponding 16 items in  $Z_n$  determined by the games in Figure 3 are given in Table 3.

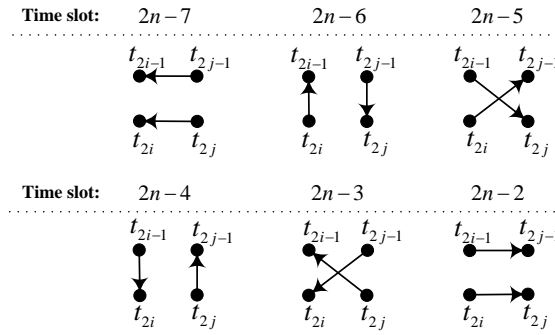
On an even time slot  $r$ , a team  $u_i$  with  $m/2 + 1 \leq i \leq m - 1$  plays against the team  $x$ . The schedule is almost the same as that in Table 3. We just need to switch the designation of home team and road team in each game. The corresponding part in  $Z_n$  is shown in Table 4.

For Case 2, we use a construction strategy different from that in Case 1 so that we are able to satisfy the condition of “at-most- $k$ ”. From this schedule, we can see that each of  $\{t_{2i-1}, t_{2i}\}$  has two away trips consisting of a single team, which are  $x_1$  and  $x_2$ , and each of  $\{x_1, x_2\}$  has an away trip consisting of two teams in  $\{t_{2i-1}, t_{2i}\}$ .

After expanding games of Cases 1 and 2, only last games on the last time slot in  $U_m$  are left unexpanded. If we expand last games according to the rules in Cases 1 and 2, superficially it will not cause trouble. However, after this there are still two games not assigned for each team, which are between two teams  $t_{2i-1}$  and  $t_{2i}$  corresponding to  $u_i$  in  $U_m$ . These two games cannot be assigned on two consecutive time slots by the “no-repeat” condition. It will be hard to find a place to schedule these two games. To solve this problem, our idea is to expand the last time slot in  $U_m$  into six (instead of four) time slots in  $Z_n$ , two of which will schedule the last two games. Then we have the following two cases.

**Case 3. Last games not involving team  $x$ :** We consider a game in  $U_m$ , where a home team  $u_i$  plays against a road team  $u_j$  in time slot  $m - 1$  ( $1 \leq i, j \leq m - 1$ ). We expand this game to  $2 \times 6 = 12$  games on six consecutive time slots, from  $2n - 7$  to  $2n - 2$ , in  $Z_n$ . Figure 4 demonstrates the 12 games on the six time slots.

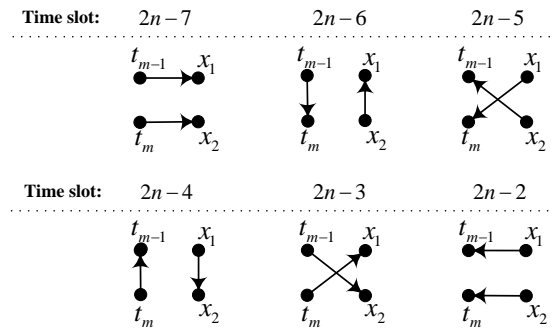
The corresponding part in  $Z_n$  is shown in Table 5.



■ **Figure 4** Expansion of Case 3.

■ **Table 5** Expanding a game of Case 3.

	$2n-7$	$2n-6$	$2n-5$	$2n-4$	$2n-3$	$2n-2$
$t_{2i-1}$	$t_{2j-1}$	$t_{2i}$	$t_{2j}$	$t_{2i}$	$t_{2j}$	$t_{2j-1}$
$t_{2i}$	$t_{2j}$	$t_{2i-1}$	$t_{2j-1}$	$t_{2i-1}$	$t_{2j-1}$	$t_{2j}$
$t_{2j-1}$	$t_{2i-1}$	$t_{2j}$	$t_{2i}$	$t_{2j}$	$t_{2i}$	$t_{2i-1}$
$t_{2j}$	$t_{2i}$	$t_{2j-1}$	$t_{2i-1}$	$t_{2j-1}$	$t_{2i-1}$	$t_{2i}$



■ **Figure 5** Expansion of Case 4.

**Case 4. Last games involving team  $x$ :** After Case 3, there is only one game in  $U_m$  left unexpanded. It is the game where team  $u_{m/2}$  plays against  $x$  on time slot  $m - 1$ . We expand this game to  $2 \times 6 = 12$  games on six consecutive time slots in  $Z_n$  according to a strategy similar to that in Figure 4. We only need to replace  $x_1$  and  $x_2$  with  $t_{2j-1}$  and  $t_{2j}$ , respectively, and switch the designation of home team and road team in each game in Figure 4. Figure 5 demonstrates the 12 games on the six time slots. The corresponding part in  $Z_n$  is given in Table 6.

It is not hard to see that the construction can be implemented in  $O(n^2)$  time. The complete tournament schedule for  $Z_8$  is given in Table 7, where time slots 1-4 and 5-8 for teams  $t_3$  and  $t_4$  correspond to Case 1, time slots 1-4 for teams  $t_1$  and  $t_2$  correspond to Case 2, time slots 9-14 for teams  $t_5$  and  $t_6$  correspond to Case 3, time slots 9-14 for teams  $t_3$  and  $t_4$  correspond to Case 4.

■ **Table 6** Expanding a game of Case 4.

	$2n-7$	$2n-6$	$2n-5$	$2n-4$	$2n-3$	$2n-2$
$t_{m-1}$	$x_1$	$t_m$	$x_2$	$t_m$	$x_2$	$x_1$
$t_m$	$x_2$	$t_{m-1}$	$x_1$	$t_{m-1}$	$x_1$	$x_2$
$x_1$	$t_{m-1}$	$x_2$	$t_m$	$x_2$	$t_m$	$t_{m-1}$
$x_2$	$t_m$	$x_1$	$t_{m-1}$	$x_1$	$t_{m-1}$	$t_m$

■ **Table 7** A Double Round-Robin Schedule for  $n = 8$ .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$t_1$	$t_7$	$t_8$	$t_7$	$t_8$	$t_3$	$t_4$	$t_3$	$t_4$	$t_5$	$t_2$	$t_6$	$t_2$	$t_6$	$t_5$
$t_2$	$t_8$	$t_7$	$t_8$	$t_7$	$t_4$	$t_3$	$t_4$	$t_3$	$t_6$	$t_1$	$t_5$	$t_1$	$t_5$	$t_6$
$t_3$	$t_5$	$t_6$	$t_5$	$t_6$	$t_1$	$t_2$	$t_1$	$t_2$	$t_7$	$t_4$	$t_8$	$t_4$	$t_8$	$t_7$
$t_4$	$t_6$	$t_5$	$t_6$	$t_5$	$t_2$	$t_1$	$t_2$	$t_1$	$t_8$	$t_3$	$t_7$	$t_3$	$t_7$	$t_8$
$t_5$	$t_3$	$t_4$	$t_3$	$t_4$	$t_7$	$t_8$	$t_7$	$t_8$	$t_1$	$t_6$	$t_2$	$t_6$	$t_2$	$t_1$
$t_6$	$t_4$	$t_3$	$t_4$	$t_3$	$t_8$	$t_7$	$t_8$	$t_7$	$t_2$	$t_5$	$t_1$	$t_5$	$t_1$	$t_2$
$t_7$	$t_1$	$t_2$	$t_1$	$t_2$	$t_5$	$t_6$	$t_5$	$t_6$	$t_3$	$t_8$	$t_4$	$t_8$	$t_4$	$t_3$
$t_8$	$t_2$	$t_1$	$t_2$	$t_1$	$t_6$	$t_5$	$t_6$	$t_5$	$t_4$	$t_7$	$t_3$	$t_7$	$t_3$	$t_4$

#### 4 Schedule based on Perfect Matchings

The above strategy provides a feasible tournament schedule for any order on the  $n$  teams. There are  $n!$  permutations of  $n$  teams. To minimize the total traveling distance, we order the teams according to a minimum perfect matching  $M$  of  $G$ , where  $G$  is a complete undirected graph on the set of teams with edge weight representing the distance between teams.

An order  $\{t_1, t_2, \dots, t_n\}$  of teams is *consistent* with a minimum perfect matching  $M$  if for each odd  $i$ , teams  $t_i$  and  $t_{i+1}$  are in a pair in  $M$ , i.e., each pair of teams corresponding to a team in  $U_m$  is also a pair in  $M$ . There are many orders of teams consistent with matching  $M$ . We will introduce a way to find a good order of teams consistent with  $M$ , which can yield tournament schedules with good performances in both theory and practice. Our algorithm contains five steps.

**Step 1.** Construct the complete undirected graph  $G$  and compute a minimum perfect matching  $M$  of  $G$ . Note that  $G$  has  $n$  vertices and  $M$  has  $n/2$  edges.

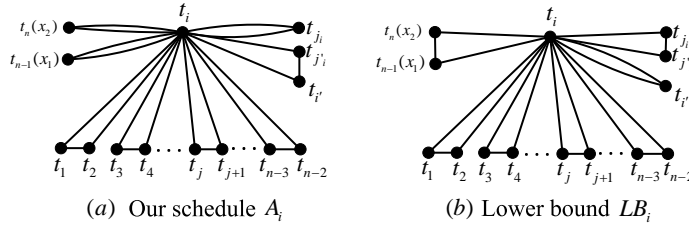
**Step 2.** Construct another complete undirected graph  $H$  based on  $G$  and  $M$ . The graph  $H$  has  $n/2$  vertices  $\{u_{i_1}, u_{i_2}, \dots, u_{i_m}\}$ . Each vertex  $u_i$  is corresponding to an edge  $t_{2i-1}t_{2i}$  in  $M$  (also a team in  $U_m$ ). The weight of each edge  $u_i u_j$  between two vertices  $u_i$  and  $u_j$  in  $H$ , denoted by  $w_H(u_i u_j)$ , is the total weight of the four edges between  $\{t_{2i-1}, t_{2i}\}$  and  $\{t_{2j-1}, t_{2j}\}$  in  $G$ .

**Step 3.** Find a vertex, denoted by  $u_m$ , in  $H$  such that the weight of all edges incident on it is minimized.

**Step 4.** Find a minimum perfect matching  $M_H$  in  $H$ . We let

$$M_H = \{u_1 u_{m-1}, u_2 u_{m-2}, \dots, u_{m/2-1} u_{m/2+1}, u_{m/2} u_m\}.$$





■ **Figure 6** Itinerary of team  $t_i$  in Case (i).

**Step 5.** Order the  $n$  teams according to  $M_H$ . We get an order  $\{t_1, t_2, \dots, t_n\}$  of the teams such that:  $t_{2i-1}$  and  $t_{2i}$  are two teams in the edge in  $M$  corresponding to  $u_i$  for  $1 \leq i \leq m$ .

This algorithm mainly computes two minimum perfect matchings and runs in  $O(n^3)$  time. From the analysis in the next section, we will see the advantages of taking this permutation of teams.

## 5 Analysis of The Approximation Ratio

We use  $A_i$  to denote the traveling distance of team  $t_i$  in our schedule. The total traveling distance in the tournament under our schedule is  $A_{ALL} = \sum_{i=1}^n A_i$ . To evaluate the performance of our schedule, we will analyze the ratio of  $A_{ALL}/LB$ . Let  $\Delta = A_{ALL} - LB$ .

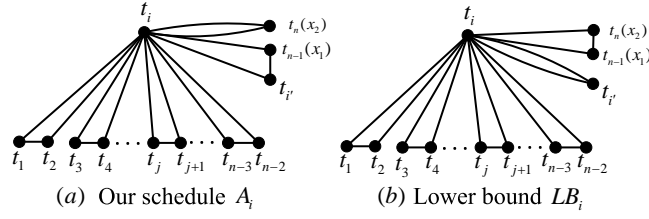
First of all, we compare  $A_i$  with the independent lower bound  $LB_i$ . Let  $\Delta_i = A_i - LB_i$ . We analyze  $\Delta_i$  for different cases according to the value of  $i$ .

**Case (i).** Teams  $t_i$  with  $1 \leq i \leq n/2 - 2$ : These teams in the double round-robin tournament  $Z_n$  are expanded from the first  $m/2 - 1$  lines (teams) in the single round-robin tournament  $U_m$ . We look at the line of team  $t_i$  in  $Z_n$ . For the part expanded from  $U_m$  of Case 1 (normal games not involving team  $x$ ),  $t_i$  has one away trip consisting of two teams in an edge in the matching  $M$ . For the part expanded from  $U_m$  of Case 2 (normal games involving team  $x$  on odd time slots),  $t_i$  has two away trips consisting of a single team of  $x_1 = t_{n-1}$  and  $x_2 = t_n$  respectively. For the part expanded from  $U_m$  of Case 3 (last games not involving team  $x$ ),  $t_i$  has an away trip consisting of a single team  $t_{j_i}$  and an away trip consisting of two teams  $t_{i'}$  and  $t_{j'_i}$ , where  $t_i$  and  $t_{i'}$  (resp.,  $t_{j_i}$  and  $t_{j'_i}$ ) correspond to the same team  $u_q$  (resp.,  $u_p$ ) in  $U_m$ , and  $p + q = m$ . The itinerary graph of team  $t_i$  is shown in Figure 6 (a). Compared to the optimal itinerary to achieve the independent lower bound, shown in Figure 6 (b), we get the following by triangle inequality

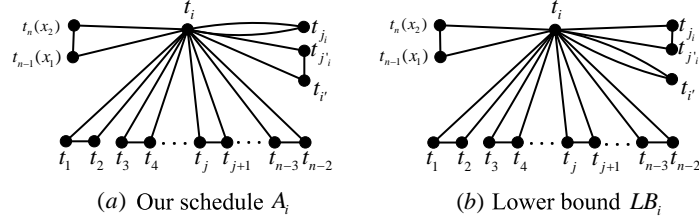
$$\begin{aligned} \Delta_i &= (D_{i,n-1} + D_{i,n}) - D_{n-1,n} \\ &\quad + (D_{i,j_i} + D_{j'_i,i'}) - (D_{i,i'} + D_{j_i,j'_i}) \\ &\leq (D_{i,n-1} + D_{i,n}) + (D_{i,j_i} + D_{i,j'_i}). \end{aligned}$$

Recall that we use  $w_H(u_i u_j)$  to denote the weight of the edge between two vertices  $u_i$  and  $u_j$  in  $H$ . We have that

$$\sum_{i=1}^{\frac{n}{2}-2} \Delta_i \leq \sum_{i=1}^{\frac{m}{2}-1} w_H(u_i u_m) + \sum_{i=1}^{\frac{m}{2}-1} w_H(u_i u_{m-i}). \quad (2)$$



■ **Figure 7** Itinerary of team  $t_i$  in Case (ii).



■ **Figure 8** Itinerary of team  $t_i$  in Case (iii).

**Case (ii).** Teams  $t_i$  with  $i \in \{n/2 - 1, n/2\}$ : These teams in  $Z_n$  are expanded from the  $\frac{m}{2}$ -th line in  $U_m$ . There are only two kinds of expansions: Case 1 and Case 4. Analogously to Case (i), we have the itinerary of  $t_{n/2-1}$  as shown in Figure 7. The itinerary graph for  $t_{n/2}$  is similar.

We get that

$$\begin{aligned} \Delta_{n/2-1} &= (D_{n/2-1,n} + D_{n/2,n-1}) \\ &\quad - (D_{n/2-1,n/2} + D_{n-1,n}) \\ &\leq D_{n/2-1,n-1} + D_{n/2-1,n} \end{aligned}$$

and also  $\Delta_{n/2} \leq D_{n/2,n-1} + D_{n/2,n}$ . Then

$$\Delta_{n/2-1} + \Delta_{n/2} \leq w_H(u_{m/2}u_m). \quad (3)$$

**Case (iii).** Teams  $t_i$  with  $n/2 + 1 \leq i \leq n - 2$ : These teams in  $Z_n$  are expanded from the lines from  $m/2 + 1$  to  $m - 1$  in  $U_m$ . There are three kinds of expansions: Case 1, Case 2 and Case 3, where the expansions of Case 2 are on even time slots. Then  $t_i$  has an away trip consisting of two teams  $x_1$  and  $x_2$ . The itinerary graph for  $t_i$  is shown in Figure 8. We get that

$$\begin{aligned} \Delta_i &= (D_{i,j_i} + D_{j'_i,i'}) - (D_{i,i'} + D_{j_i,j'_i}) \\ &\leq D_{i,j_i} + D_{i,j'_i}, \end{aligned}$$

where  $t_i$  and  $t_{i'}$  (resp.,  $t_{j_i}$  and  $t_{j'_i}$ ) correspond to the same team  $u_q$  (resp.,  $u_p$ ) in  $U_m$ , and  $p + q = m$ .

By summing up  $i$ 's in this case, we get

$$\sum_{i=n/2+1}^{n-2} \Delta_i \leq \sum_{i=m/2+1}^{m-1} w_H(u_i u_{m-i}). \quad (4)$$

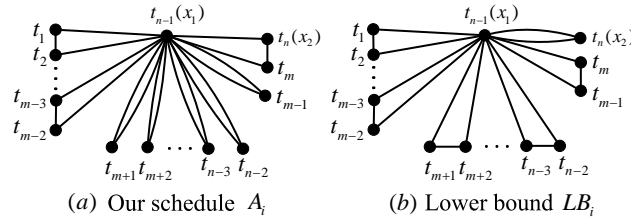


Figure 9 Itinerary of team  $x_1$  in Case (iv).

**Case (iv).** Teams  $t_i$  with  $i \in \{n-1, n\}$ : The last two teams in  $Z_n$  are expanded from the last line in  $U_m$ . There are two kinds of expansions involving  $x$ : Case 2 and Case 4. The itinerary graph for  $x_1 = t_{n-1}$  is shown in Figure 9.

We get that

$$\begin{aligned} \Delta_{n-1} &= \sum_{j=m/2+1}^{m-1} ((D_{n-1,2j-1} + D_{n-1,2j}) - D_{2j-1,2j}) \\ &\quad + (D_{m-1,n-1} + D_{m,n}) - (D_{m-1,m} + D_{n-1,n}) \\ &\leq \sum_{j=m/2+1}^{m-1} (D_{n-1,2j-1} + D_{n-1,2j}) \\ &\quad + (D_{m,n-1} + D_{m-1,n-1}) \\ &= \sum_{j=m/2}^{m-1} (D_{n-1,2j-1} + D_{n-1,2j}) \end{aligned}$$

and also  $\Delta_n \leq \sum_{j=m/2}^{m-1} (D_{n,2j-1} + D_{n,2j})$ . Then

$$\Delta_{n-1} + \Delta_n \leq \sum_{j=m/2}^{m-1} w_H(u_m u_j). \tag{5}$$

By summing up (2), (3), (4) and (5), we get

$$\begin{aligned} \Delta &= \sum_{i=1}^n \Delta_i \leq \sum_{j=1}^{m-1} w_H(u_m u_j) \\ &\quad + 2 \sum_{i=1}^{\frac{m}{2}-1} w_H(u_i u_{m-i}) + w_H(u_{m/2} u_m) \\ &\leq w_H(E(u_m)) + 2w_H(M_H), \end{aligned} \tag{6}$$

where  $w_H(E(u_m))$  is the total weight of all edges incident on  $u_m$  in  $H$  and  $w_H(M_H)$  is weight of all edges in the matching  $M_H$ . Let  $D_H$  denote the weight of all edges in  $H$ . Then

$$D_H = D_E - D_M. \tag{7}$$

Since we select  $u_m$  as the vertex in  $H$  such that the weight of all edges incident on it is minimized, we know that

$$w_H(E(u_m)) \leq \frac{2}{m} D_H. \tag{8}$$

Note that a complete graph of  $m$  vertices can be partitioned into  $m-1$  perfect matchings. We select  $M_H$  as a perfect matching of minimum weight. Then we have that

$$w_H(M_H) \leq \frac{1}{m-1} D_H. \tag{9}$$

By (1), (6), (7), (8), (9) and  $n = 2m$ , we get

$$\Delta \leq \left(\frac{2}{m} + \frac{2}{m-1}\right) D_H \leq \left(\frac{2}{n} + \frac{2}{n-2}\right) LB,$$

which implies

► **Theorem 1.** For TTP-2 with  $n$  teams such that  $n \equiv 0 \pmod{4}$ , the above algorithm runs in  $O(n^3)$  time and finds a feasible schedule such that the traveling distance is at most  $1 + \frac{2}{n-2} + \frac{2}{n}$  times of the optimal traveling distance.

■ **Table 8** The results for real-world instances.

Data set	Lower bounds	Previous results	Before search	After search	Our gap (%)
Galaxy40	298484	318033	308235	307469	3.01
Galaxy36	205280	220537	213160	212821	3.67
Galaxy32	139922	148395	145857	145445	3.95
Galaxy28	89242	94389	93317	93235	4.47
Galaxy24	53282	56476	55959	55883	4.88
Galaxy20	30508	33211	32548	32530	6.63
Galaxy16	17562	19432	19124	19040	8.42
Galaxy12	8374	9570	9546	9490	13.33
NFL32	1162798	1268742	1212521	1211239	4.17
NFL28	771442	832396	811586	810310	5.04
NFL24	573618	641686	612928	611441	6.59
NFL20	423958	485618	458099	456563	7.69
NFL16	294866	332468	322528	321357	8.98
NL16	334940	380179	360207	359720	7.40
NL12	132720	148382	145035	144744	9.06
super12	551580	680054	613107	612583	11.06
brazil24	620574	722281	655603	655235	5.59

## 6 Local Search by Swapping

Some simple local search techniques can still be applied to our schedule. These techniques may not be able to improve approximation ratio in theory. However, in practice, for most benchmark instances they still can slightly improve our results by about 1%. We use only two simple search rules:

- Swap two pairs of teams in the matching  $M_H$ ;
- Swap any pair of teams.

## 7 Applications to Benchmark Sets

To show the efficiency of our algorithm in practice, we apply it to the benchmark instances provided on the website of Trick [19], most of which are real-world instances. The website of Trick [19] displays the best results to TTP-3 on these instances, while we focus on the results to TTP-2. Table 8 lists our results and the best-known results [18] for all 17 instances with  $n \equiv 0 \pmod{4}$  and  $n > 8$ , where “lower bound” is the independent lower bound, “before search” and “after search” mean our results before and after applying local search by swapping respectively, and “our gap” is defined to be  $\Delta/LB$ . Our results beat all previously best-known upper bounds, most by about 3% to 10%. It is also worthy to note that our algorithm computes all instances together within 2.6 seconds on a standard laptop with a 2.40GHz Intel(R) Core(TM) i5-2430 CPU and 4 gigabytes of memory.

## 8 Conclusion

Our tournament schedule generates a feasible solution to TTP-2 with  $n \equiv 0 \pmod{4}$ . Our solution is at most  $1 + \frac{2}{n-2} + \frac{2}{n}$  times of the optimal, improving the previous approximation

ratio of  $1 + \frac{16}{n}$  by an addition of almost  $\frac{12}{n}$ . By applying our algorithm on several benchmark sets of TTP, our tournament schedules beat best-known solutions for all instances with  $n \equiv 0 \pmod{4}$ .

The number  $n$  of teams in TTP is required to be even. When we construct a double round-robin tournament from a single round-robin tournament, we further require that the number  $m = n/2$  of teams in the single round-robin tournament is even. Thus, our constructive algorithm requires  $n \equiv 0 \pmod{4}$ . The only left case not considered in this paper is  $n \equiv 2 \pmod{4}$ . For this case, the previously-known approximation ratio is  $\frac{3}{2} + \frac{6}{n-4}$  [18], and the gaps between upper and lower bounds on benchmark instances are large. A natural question is whether there is a  $(1 + O(1/n))$ -approximation algorithm for the case that  $n \equiv 2 \pmod{4}$ .

---

## References

- 1 Aris Anagnostopoulos, Laurent Michel, Pascal Van Hentenryck, and Yannis Vergados. A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2):177–193, 2006.
- 2 Rishiraj Bhattacharyya. A note on complexity of traveling tournament problem. *Optimization Online*, 2009.
- 3 Robert Thomas Campbell and DS Chen. A minimum distance basketball scheduling problem. *Management science in sports*, 4:15–26, 1976.
- 4 Dominique de Werra. Some models of graphs for scheduling sports competitions. *Discrete Applied Mathematics*, 21(1):47–65, 1988.
- 5 Luca Di Gaspero and Andrea Schaerf. A composite-neighborhood tabu search approach to the traveling tournament problem. *Journal of Heuristics*, 13(2):189–207, 2007.
- 6 Kelly Easton, George Nemhauser, and Michael Trick. The traveling tournament problem: description and benchmarks. In *7th International Conference on Principles and Practice of Constraint Programming*, pages 580–584, 2001.
- 7 Kelly Easton, George Nemhauser, and Michael Trick. Solving the travelling tournament problem: a combined integer programming and constraint programming approach. In *4th International Conference of Practice and Theory of Automated Timetabling IV*, pages 100–109, 2003.
- 8 Nobutomo Fujiwara, Shinji Imahori, Tomomi Matsui, and Ryuhei Miyashiro. Constructive algorithms for the constant distance traveling tournament problem. *Practice and Theory of Automated Timetabling VI*, pages 135–146, 2007.
- 9 Marc Goerigk, Richard Hoshino, Ken Kawarabayashi, and Stephan Westphal. Solving the traveling tournament problem by packing three-vertex paths. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2271–2277, 2014.
- 10 Richard Hoshino and Ken Kawarabayashi. The linear distance traveling tournament problem. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1770–1778, 2012.
- 11 Richard Hoshino and Ken-ichi Kawarabayashi. An approximation algorithm for the bipartite traveling tournament problem. *Mathematics of Operations Research*, 38(4):720–728, 2013.
- 12 Shinji Imahori, Tomomi Matsui, and Ryuhei Miyashiro. A 2.75-approximation algorithm for the unconstrained traveling tournament problem. *Annals of Operations Research*, 218(1):237–247, 2014.
- 13 Graham Kendall, Sigrid Knust, Celso C Ribeiro, and Sebastián Urrutia. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, 37(1):1–19, 2010.
- 14 Andrew Lim, Brian Rodrigues, and X Zhang. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3):1459–1478, 2006.

- 15    Ryuhei Miyashiro, Tomomi Matsui, and Shinji Imahori. An approximation algorithm for the traveling tournament problem. *Annals of Operations Research*, 194(1):317–324, 2012.
- 16    Rasmus V Rasmussen and Michael A Trick. Round robin scheduling—a survey. *European Journal of Operational Research*, 188(3):617–636, 2008.
- 17    Clemens Thielen and Stephan Westphal. Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4):345–351, 2011.
- 18    Clemens Thielen and Stephan Westphal. Approximation algorithms for TTP(2). *Mathematical Methods of Operations Research*, 76(1):1–20, 2012.
- 19    Michael Trick. Challenge traveling tournament instances. Online reference at <http://mat.gsia.cmu.edu/TOURN/>, 2013.
- 20    Pascal Van Hentenryck and Yannis Vergados. Population-based simulated annealing for traveling tournaments. In *Twenty-Second AAAI Conference on Artificial Intelligence*, pages 267–272, 2007.
- 21    Stephan Westphal and Karl Noparlik. A 5.875-approximation for the traveling tournament problem. *Annals of Operations Research*, 218(1):347–360, 2014.
- 22    Daisuke Yamaguchi, Shinji Imahori, Ryuhei Miyashiro, and Tomomi Matsui. An improved approximation algorithm for the traveling tournament problem. *Algorithmica*, 61(4):1077–1091, 2011.