The Linear Distance Traveling Tournament Problem Allows an EPTAS

Jingyang Zhao, Mingyu Xiao*

University of Electronic Science and Technology of China
jingyangzhao1020@gmail.com, myxiao@gmail.com

Abstract

The Traveling Tournament Problem (TTP-k) is a well-known benchmark problem in tournament timetabling and has been extensively studied in the field of AI. In this problem, we are going to design a double round-robin schedule such that each pair of teams plays one game in each other's home venue, minimizing the total distance traveled by all n teams (n is even) under the constraint that each team can have at most k-consecutive home games or away games. The Linear Distance Traveling Tournament Problem (LDTTP-k), where all teams are located on a line, was introduced by Hoshino and Kawarabayashi (AAAI 2012). For LDTTP-3, they gave a $4/3$-approximation algorithm for $n \equiv 4 \pmod 6$ teams. In this paper, we show that for any $3 \leq k = o(\sqrt[3]{n})$, LDTTP-$k$ allows an efficient polynomial-time approximation scheme (EPTAS).

Introduction

The traveling tournament problem (TTP-k), systematically introduced in (Easton, Nemhauser, and Trick 2001), is a widely studied benchmark problem in the field of sports schedules (Kendall et al. 2010). This problem aims to find a feasible double round-robin tournament under some constraints such that the total traveling distance of all participant teams is minimized. A double round-robin tournament of n teams will last $2(n-1)$-consecutive days, and each team plays exactly one game against another team each day. Each team plays two games against each of the other $n-1$ teams: one home game at its home venue and one away game at its opponent's home venue. For TTP-k, there are two more constraints on the round-robin tournament:

- *No-repeat*: No pair of teams can play against each other in two consecutive games.
- *Bounded-by-k*: Each team can play at most k-consecutive home games or away games.

In the problem, we also assume that each team travels directly from its game venue on i-th day to its venue on $(i+1)$-th day, and each team is at home before the first game starts and will return home after the last game ends. We may also assume that the distance is a metric, i.e., it satisfies the symmetry and triangle inequality properties.

The linear distance traveling tournament problem (LDTTP-k), introduced in (Hoshino and Kawarabayashi 2012), is an interesting variant of TTP-k where all teams are located on a line.

Related Work

TTP-k has been extensively studied both in theory and practice. TTP-k is known to be NP-hard for any fixed $k \geq 3$ and $k = n-1$ (Thielen and Westphal 2011; Chatterjee 2021; Bhattacharyya 2016). The smaller the value of k, the more frequently each team has to return home. The case of $k = 3$ is the most extensively studied. However, it is not easy to find a feasible solution, and many instances of TTP-3 with more than ten teams in the online benchmark (Trick 2020; Bulck et al. 2020) have not been completely solved even by using high-performance machines.

In theory, most contributions are about approximation algorithms (Thielen and Westphal 2012; Xiao and Kou 2016; Chatterjee and Roy 2021; Imahori 2021; Miyashiro, Matsui, and Imahori 2012; Yamaguchi et al. 2011; Westphal and Noparlik 2014; Hoshino and Kawarabayashi 2012, 2013). Now, the approximation ratios have been improved to $1+\varepsilon$ for TTP-2 (Zhao and Xiao 2021a,b), $139/87+\varepsilon$ for TTP-3 (Zhao, Xiao, and Xu 2022), and $11/4$ for TTP-$(n-1)$ (Imahori, Matsui, and Miyashiro 2014). There is also a large number of works on practical and heuristic algorithms (Easton, Nemhauser, and Trick 2002; Lim, Rodrigues, and Zhang 2006; Anagnostopoulos et al. 2006; Di Gaspero and Schaerf 2007; Hentenryck and Vergados 2007; Goerigk et al. 2014; Goerigk and Westphal 2016).

LDTTP-k is a linear distance relaxation of TTP-k, where all teams are on a line. It reduces the $\frac{n(n-1)}{2}$ pairwise distance parameters to just $n-1$ variables. Although the hardness of LDTTP-k is still not formally proved yet, it is believed that LDTTP-k is also NP-hard since it is not even easy to construct a feasible solution. For LDTTP-3, Hoshino and Kawarabayashi (2012) proposed a $4/3$-approximation algorithm for the case of $n \equiv 4 \pmod 6$. Whether there exist similar constructions for $n \equiv 0$ and $n \equiv 2 \pmod 6$ was asked in (Hoshino and Kawarabayashi 2012; Goerigk et al. 2014). Very recently, the approximation ratio for LDTTP-3 was improved to $6/5+\varepsilon$ (Zhao, Xiao, and Xu 2022).

*Corresponding author

Our Results

In this paper, we study algorithms for LDTTP-k. We show that for any $3 \leq k = o(\sqrt[3]{n})$, LDTTP-$k$ allows an EPTAS: for any constant $\varepsilon > 0$, we can compute a schedule for LDTTP-k with the weight at most $1 + \varepsilon$ times the optimal in time $O_\varepsilon(1) + n^{O(1)}$. Our result significantly improves all previously known results. In this paper, we mainly focus on theoretical analysis of the approximation ratio.

For the sake of analysis, we adopt some simple randomized methods in our schedule, which can be derandomized efficiently by using the classic method of conditional expectations. Due to limited space, the proofs of lemmas and theorems marked with "*" were omitted and they can be found in the full version of this paper.

Notations

Let n denote the number of total teams, where n is always an even number by the definition of the problem. Recall that k is the maximum number of consecutive home/away games allowed. In our setting, k is a part of the input. However, we assume $3 \leq k = o(\sqrt[3]{n})$ since our schedule and analysis require this condition, which already includes the case that k is a constant. We use $G = (V, E)$ to denote the complete graph on the n vertices representing the n teams. There is a weight function $w : E \to \mathbb{R}_{\geq 0}$ on the edges of G. The weight $w(u, v)$ of edge uv is the distance between the homes of teams u and v. For a subset of edges $E' \subseteq E$, we define $w(E') = \sum_{x \in E'} w(x)$.

We use $E(u)$ to denote the set of edges incident on u in G. The *weighted degree* of a vertex u is defined as $\delta(u) = w(E(u))$. We let Δ be the sum of the weighted degrees of all vertices, i.e.,

$$\Delta = \sum_{u \in V} \delta(u) = 2w(E).$$

A *k-path P* is a simple path on k different vertices. A *k-path packing* $\mathcal{P}$ in G is a set of edges such that every component is a k-path, and each vertex is covered. A k-path on k vertices $v_1 \dots v_k$ in this order is denoted by $\{v_1, \dots, v_k\}$. The two vertices v_1 and v_k are called *terminals* of the k-path.

When n is not divisible by k, there is no k-path packing in G. For this case, we may consider an induced subgraph G_ε of G with the number of vertices being a multiple of k. We use n_ε to denote the number of vertices in G_ε, and define the corresponding notations $E_\varepsilon(\cdot)$, $\delta_\varepsilon(\cdot)$ and Δ_ε in G_ε.

Lower Bounds

We first present two lower bounds for LDTTP-k, which will be used to compare with our solution. We use OPT to denote the value of an optimal solution for LDTTP-k or TTP-k.

Since LDTTP-k is a special case of TTP-k, all lower bounds for TTP-k also hold for LDTTP-k.

The first lower bound is a known bound for TTP-k, which can be obtained by the triangle inequality property.

Lemma 1. (Westphal and Noparlik 2014). *For TTP-k, it holds that $\Delta \leq (k/2) \cdot OPT$.*

The second lower bound uses the nature of LDTTP-k. We consider the induced graph $G_\varepsilon = G[V_\varepsilon]$ which is also a line shape. There are $n_\varepsilon = |V_\varepsilon|$ teams in G_ε. We label them with $t_1, t_2, \dots, t_{n_\varepsilon}$ from left to right on the line respectively, and let $d_i = w(t_i, t_{i+1})$ for $1 \leq i \leq n_\varepsilon - 1$. We have

Lemma 2. *For LDTTP-k, it holds that $\sum_{i=1}^{n_\varepsilon - 1} l_i d_i \leq OPT$, where $l_i = 2i\lceil \frac{n_\varepsilon - i}{k} \rceil + 2(n_\varepsilon - i)\lceil \frac{i}{k} \rceil$.*

Proof. For an edge $t_i t_{i+1}$, there are i teams on the left and $n_\varepsilon - i$ teams on the right of it. For each left team, it takes at least $\lceil \frac{n_\varepsilon - i}{k} \rceil$ trips for it to visit all right teams. For each right team, it takes at least $\lceil \frac{i}{k} \rceil$ trips for it to visit all left teams. There are at least $i\lceil \frac{n_\varepsilon - i}{k} \rceil + (n_\varepsilon - i)\lceil \frac{i}{k} \rceil = \frac{l_i}{2}$ trips in total. Each trip must cross the edge twice and hence all teams must cross the edge at least l_i times. Together, we get the lower bound $\sum_{i=1}^{n_\varepsilon - 1} l_i d_i$. □

The lower bound in Lemma 2 with $k = 3$ can be used to get the $4/3$-approximation algorithm for the case $n \equiv 4 \pmod 6$ by (Hoshino and Kawarabayashi 2012). It is not enough to get a PTAS. We reveal more properties.

Note that n_ε is divisible by k. We can construct a k-path packing $\mathcal{P}_\varepsilon^* = \{t_{ki-k+1} t_{ki-k+2} \dots t_{ki}\}_{i=1}^{n_\varepsilon/k}$ by packing the k-paths from the most left to the most right along the line. It is easy to see that $\mathcal{P}_\varepsilon^*$ is the optimal k-path packing with the minimum weight of the paths in it. We define $\Delta_\varepsilon(\mathcal{P}_\varepsilon^*)$ as the sum weighted degrees of the terminals in $\mathcal{P}_\varepsilon^*$. That is

$$\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) = \sum_{i=1}^{n_\varepsilon/k} (\delta_\varepsilon(t_{ki-k+1}) + \delta_\varepsilon(t_{ki})).$$

If each team in G_ε plays k-consecutive away games along each k-path in $\mathcal{P}_\varepsilon^*$ from one terminal to another terminal, then the total traveling distance of all teams is exactly $\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) + n_\varepsilon w(\mathcal{P}_\varepsilon^*)$. Interestingly, we have

Lemma 3 (*). $\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) + n_\varepsilon w(\mathcal{P}_\varepsilon^*) = \sum_{i=1}^{n_\varepsilon - 1} l_i d_i$.

By Lemmas 2 and 3, we get that

Lemma 4. *For LDTTP-k, it holds that $\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) + n_\varepsilon w(\mathcal{P}_\varepsilon^*) \leq OPT$.*

Lemma 4 shows that to obtain a solution close to the optimal (or PTAS), almost all teams in G_ε may need to play k-consecutive away games with the teams in every k-path in $\mathcal{P}_\varepsilon^*$. Next, we will introduce our construction.

The Construction of the Schedule

For any constant $\varepsilon > 0$, we will construct a schedule with the total weight at most $1 + \varepsilon$ times the optimal. Hence, our construction depends on the value ε, i.e., our schedule may be different for different values of ε. Let $d = 12\lceil 1/\varepsilon \rceil$ (d is an even constant). Then, we can get $d = \Theta(1/\varepsilon) = O_\varepsilon(1)$. We also let $m = 2\lfloor \frac{n}{2kd} \rfloor$. So, m is also an even number.

Then, we can use $O(n^2)$ time to choose mkd vertices in V to form the vertex set V_ε such that the sum weighted degrees of them $\sum_{t_i \in V_\varepsilon} \delta(t_i)$ is maximized. We can get

$$\sum_{t_i \in V_\varepsilon} \delta(t_i) \geq (mkd/n) \sum_{t_i \in V} \delta(t_i) = (mkd/n)\Delta. \qquad (1)$$

The number of vertices in V_ε is divisible by k. The graph G_ε induced by V_ε has an optimal k-path packing $\mathcal{P}^*_\varepsilon$ (defined above). Our construction is based on the k-path packing $\mathcal{P}^*_\varepsilon$. We aim to construct a feasible schedule such that each team plays k-consecutive away games against teams in each k-path in the packing as much as possible. More precisely, the weight of our schedule is bounded by

$$(1+O(\varepsilon))(\varDelta_\varepsilon(\mathcal{P}^*_\varepsilon)+n_\varepsilon w(\mathcal{P}^*_\varepsilon))+O(\varepsilon)\cdot \mathrm{OPT}.$$

By Lemma 4, we can get an approximation ratio of $1+O(\varepsilon)$ for LDTTP-k.

We also assume $n \geq 2k^2d^2$. Otherwise, since $k=o(\sqrt[3]{n})$ and $d=O_\varepsilon(1)$, we know that there is a constant $n_0=O_\varepsilon(1)$ such that $n\leq n_0$. We can use a brute-and-force algorithm to solve the problem in constant time $O_\varepsilon(1)$. Note that we can get $m=2\lfloor\frac{n}{2kd}\rfloor \geq 2kd$. Next, we are going to introduce the main framework of the construction.

To make our construction clear, we first partition teams and the days of games into several groups and design a higher-level schedule for games among groups.

Each team should attend $2(n-1)$ games on $2(n-1)$ days. We partition the $2(n-1)$ days into $m-1$ *time slots*. Each of the first $m-2$ time slots lasts $2kd$ days and the last time slot contains the remaining $2n-2-2kd(m-2)$ days.

We also partition the n teams into different groups. First, we partition the set V into two parts: V_ε and $V\setminus V_\varepsilon$. There are $n_\varepsilon=|V_\varepsilon|=mkd$ teams in V_ε. We find an optimal k-path packing $\mathcal{P}^*_\varepsilon$ in G_ε. The size of $\mathcal{P}^*_\varepsilon$ is md. We randomly label the k-paths in $\mathcal{P}^*_\varepsilon$ by $\{P_1,\ldots,P_{md}\}$ and take each k-path P_i as a *path-team* u_i. We regard u_i as a set of the k teams in the corresponding k-path and relabel $u_i=\{t_{ki-k+1},\ldots,t_{ki}\}$ for the sake of presentation. We also consider d path-teams as a *super-team* U_i: $U_i=\{u_{di-d+1},\ldots,u_{di}\}$. Thus, we will get m super-teams $\{U_1,\ldots,U_m\}$. There are still $n-mkd$ teams in $V\setminus V_\varepsilon$. We arbitrarily partition them into $r=n/2-mkd/2$ *team-pairs*, denoted by $\{R_1,\ldots,R_r\}$, where we relabel $R_i=\{t_{n_\varepsilon+2i-1},t_{n_\varepsilon+2i}\}$ for the sake of presentation. Note that $r\leq kd-1$.

The main idea of the construction is that: we first arrange a schedule of *super-games* between super-teams (including the team-pairs); then, we extend the super-games into normal games between normal teams, which will form a feasible schedule for LDTTP-k.

For the schedule of super-games, there are m super-teams, and each of them will attend $m-1$ super-games in $m-1$ consecutive time slots. In each of the first $m-2$ time slots, we have $m/2$ super-games: 1 is called the *left super-game*, involving two super-teams; $m/2-r-1$ are called the *middle super-games*, each of them involving two super teams; r are called the *right super-games*, each of them involving two super teams and one team-pair. Note that $m/2\geq r+1$ since $r\leq kd-1$ and $m\geq 2kd$ by the previous settings. The last time slot is special, and we will explain it later.

The First $m-2$ Time Slots

In the first time slot, the $m/2$ super-games are arranged as shown in Figure 1 (for $m=10$ and $r=2$). The last super-team U_m is denoted by a double-cycle node on the left, the other $m-1$ super-teams, denoted by single-cycle nodes, form a cycle, and the r team-pairs are denoted by r square nodes on the right. Each super-game is denoted by a directed edge between two super-teams. The information on the direction will be used later. We regard that the super-game happens at the home of U_j if the direction is from U_i to U_j. The most left super-game involving super-team U_m is called the *left super-game* and we put a letter 'L' on edge to denote it. The middle $m/2-1-r$ super-games are called *middle super-games*. Note that the direction of the arcs is alternative. The most right r super-games, each involving two super-teams and one team-pair, are called the *right super-games*. For the sake of the presentation, we put the square node for a team-pair beside the edge.

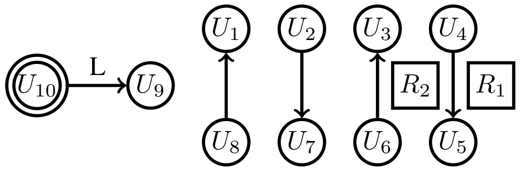
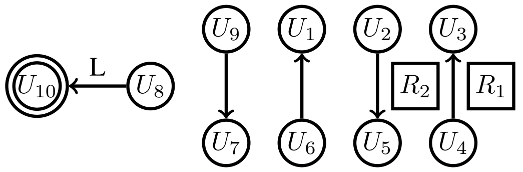


Figure 1: The super-game schedule in the first time slot, where $m=10$ and $r=2$



$$\begin{tikzcd} U_{10} & U_8 \arrow[l, "L"'] \end{tikzcd}$$
U_9
U_7
U_1
U_6
U_2
U_5
R_2
U_3
U_4
R_1

Figure 2: The super-game schedule in the second time slot, where $m=10$ and $r=2$

In the second time slot, super-games are scheduled as shown in Figure 2. We change the positions of single-cycle nodes in the cycle $U_1U_2\ldots U_{m-1}$ by moving one position in the clockwise direction, and also change the direction of each edge. The positions of the double-cycle node U_m and the r square nodes are fixed.

The schedules for the first $m-2$ time slots are derived analogously. There are three kinds of super-games in the first $m-2$ time slots: left, middle, and right. We explain how to extend them into normal games one by one. We first consider the easy case of middle super-games.

Middle super-games: We consider a middle super-game from super-teams U_i to U_j. Recall that each super-team contains d path-teams. We will first extend the middle super-game into *path-games* between path-teams $u_{i'}\in U_i$ and $u_{j'}\in U_j$. Then, we will further extend the path-games into normal games between normal teams. The time slot is split into d *sessions*. Each path-team $u_{i'}\in U_i$ will play an away path-game with each path-team $u_{j'}\in U_j$ in a session. In Table 1, we show the path-games for $d=6$, where we relabel $U_i=\{u_1,\ldots,u_6\}$ and $U_j=\{v_1,\ldots,v_6\}$ to make it neat.

Next, we extend path-games into normal games. Consider a path-game from path-teams u_i to v_j $(1\leq i,j\leq d)$. Let $u_i=\{x_{ki-k},\ldots,x_{ki-1}\}$ and $v_j=\{y_{kj-k},\ldots,y_{kj-1}\}$. In a session, which contains $2k$ days, each team $x_{ki-k+k'}\in u_i$ (resp., $y_{kj-k+k'}\in v_j$) plays k'-consecutive home

	1	2	3	4	5	6
u_1	v_1	v_2	v_3	v_4	v_5	v_6
u_2	v_6	v_1	v_2	v_3	v_4	v_5
u_3	v_5	v_6	v_1	v_2	v_3	v_4
u_4	v_4	v_5	v_6	v_1	v_2	v_3
u_5	v_3	v_4	v_5	v_6	v_1	v_2
u_6	v_2	v_3	v_4	v_5	v_6	v_1
v_1	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$
v_2	$\boldsymbol{u_6}$	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$
v_3	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$
v_4	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$
v_5	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$
v_6	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$	$\boldsymbol{u_1}$

Table 1: Extending the middle super-game from $U_i = \{u_1, \dots, u_6\}$ to $U_j = \{v_1, \dots, v_6\}$ into path-games in $d = 6$ sessions, where home path-games are marked in bold

	1	2	3	4	5	6
x_{3i-3}	y_{3j-3}	y_{3j-2}	y_{3j-1}	$\boldsymbol{y_{3j-3}}$	$\boldsymbol{y_{3j-2}}$	$\boldsymbol{y_{3j-1}}$
x_{3i-2}	$\boldsymbol{y_{3j-4}}$	y_{3j-3}	y_{3j-2}	y_{3j-1}	$\boldsymbol{y_{3j-3}}$	$\boldsymbol{y_{3j-2}}$
x_{3i-1}	$\boldsymbol{y_{3j-5}}$	$\boldsymbol{y_{3j-4}}$	y_{3j-3}	y_{3j-2}	y_{3j-1}	$\boldsymbol{y_{3j-3}}$
y_{3j-3}	$\boldsymbol{x_{3i-3}}$	$\boldsymbol{x_{3i-2}}$	$\boldsymbol{x_{3i-1}}$	x_{3i-3}	x_{3i-2}	x_{3i-1}
y_{3j-2}	x_{3i-4}	$\boldsymbol{x_{3i-3}}$	$\boldsymbol{x_{3i-2}}$	$\boldsymbol{x_{3i-1}}$	x_{3i-3}	x_{3i-2}
y_{3j-1}	x_{3i-5}	x_{3i-4}	$\boldsymbol{x_{3i-3}}$	$\boldsymbol{x_{3i-2}}$	$\boldsymbol{x_{3i-1}}$	x_{3i-3}

Table 2: Extending the path-game from $u_i = \{x_{3i-3}, x_{3i-2}, x_{3i-1}\}$ to $v_j = \{y_{3j-3}, y_{3j-2}, y_{3j-1}\}$ into normal games in $2k = 6$ days, where home games are marked in bold

(resp., away) games with k' teams in v_{j-1} (resp., u_{i-1}), k-consecutive away (resp., home) games with k teams in v_j (resp., u_i), and $(k - k')$-consecutive home (resp., away) games with $k - k'$ teams in v_j (resp., u_i) in $2k$ days ($0 \leq k' \leq k - 1$). The schedule is shown in Table 2 for $k = 3$. Here we also regard $u_0 = u_d$ (resp., $v_0 = v_d$).

The design of normal super-games is inspired by the algorithm in (Hoshino and Kawarabayashi 2013). From Table 2, we can see that in the path-game, every normal team in u_i will play k-consecutive away games along the k-path of v_j while the normal team in v_j may not. However, each path-team in U_j plays d-consecutive home path-games, as shown in Table 1. By combining all path-games together, we can see that every normal team in U_i will play d away trips along the d k-paths in U_j, and every normal team in U_j will play d or $d + 1$ away trips along at least $d - 1$ k-paths in U_i. Furthermore, all normal games are arranged between one normal team in U_i and one normal team in U_j. Roughly speaking, the bigger the value d, the more successful the schedule. Later, we will show that it is good enough to set $d = \Theta(1/\varepsilon)$.

An illustration of the normal games after extending one middle super-game for $k = 3$ and $d = 2$ is shown in Table 3.

Left super-games: We consider a left super-game from super-teams U_i to U_j. Note that one of U_i and U_j is U_m. Let $U_i = \{\{x_0, \dots, x_{k-1}\}, \dots, \{x_{kd-k}, \dots, x_{kd-1}\}\}$ and $U_j = \{\{y_0, \dots, y_{k-1}\}, \dots, \{y_{kd-k}, \dots, y_{kd-1}\}\}$. For all

	1	2	3	4	5	6	7	8	9	10	11	12
x_0	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	y_3	y_4	y_5	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$
x_1	$\boldsymbol{y_5}$	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	y_3	y_4	y_5	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$
x_2	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	y_3	y_4	y_5	$\boldsymbol{y_3}$
x_3	y_3	y_4	y_5	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$
x_4	$\boldsymbol{y_2}$	y_3	y_4	y_5	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$
x_5	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	y_3	y_4	y_5	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$	y_0	y_1	y_2	$\boldsymbol{y_0}$
y_0	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1	x_2	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3	x_4	x_5
y_1	x_5	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1	x_2	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3	x_4
y_2	x_4	x_5	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1	x_2	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3
y_3	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3	x_4	x_5	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1	x_2
y_4	x_2	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3	x_4	x_5	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1
y_5	x_1	x_2	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	x_3	x_4	x_5	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0

Table 3: Extending the middle super-game from $U_i = \{\{x_0, x_1, x_2\}, \{x_3, x_4, x_5\}\}$ to $U_j = \{\{y_0, y_1, y_2\}, \{y_3, y_4, y_5\}\}$ into normal games in $2kd = 12$ days, where home normal games are marked in bold

	1	2	3	4	5	6	7	8	9	10	11	12
x_0	y_0	$\boldsymbol{y_1}$	y_2	$\boldsymbol{y_3}$	y_4	$\boldsymbol{y_5}$	$\boldsymbol{y_0}$	y_1	$\boldsymbol{y_2}$	y_3	$\boldsymbol{y_4}$	y_5
x_1	y_5	$\boldsymbol{y_0}$	y_1	$\boldsymbol{y_2}$	y_3	$\boldsymbol{y_4}$	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_1}$	y_2	$\boldsymbol{y_3}$	y_4
x_2	y_4	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_1}$	y_2	$\boldsymbol{y_3}$	$\boldsymbol{y_4}$	y_5	$\boldsymbol{y_0}$	y_1	$\boldsymbol{y_2}$	y_3
x_3	y_3	$\boldsymbol{y_4}$	y_5	$\boldsymbol{y_0}$	y_1	$\boldsymbol{y_2}$	$\boldsymbol{y_3}$	y_4	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_1}$	y_2
x_4	y_2	$\boldsymbol{y_3}$	y_4	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	y_3	$\boldsymbol{y_4}$	y_5	$\boldsymbol{y_0}$	y_1
x_5	y_1	$\boldsymbol{y_2}$	y_3	$\boldsymbol{y_4}$	y_5	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	y_2	$\boldsymbol{y_3}$	y_4	$\boldsymbol{y_5}$	y_0
y_0	$\boldsymbol{x_0}$	x_1	$\boldsymbol{x_2}$	x_3	$\boldsymbol{x_4}$	x_5	x_0	$\boldsymbol{x_1}$	x_2	$\boldsymbol{x_3}$	x_4	$\boldsymbol{x_5}$
y_1	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_1}$	x_2	$\boldsymbol{x_3}$	x_4	x_5	$\boldsymbol{x_0}$	x_1	$\boldsymbol{x_2}$	x_3	$\boldsymbol{x_4}$
y_2	$\boldsymbol{x_4}$	x_5	$\boldsymbol{x_0}$	x_1	$\boldsymbol{x_2}$	x_3	x_4	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_1}$	x_2	$\boldsymbol{x_3}$
y_3	$\boldsymbol{x_3}$	x_4	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_1}$	x_2	x_3	$\boldsymbol{x_4}$	x_5	$\boldsymbol{x_0}$	x_1	$\boldsymbol{x_2}$
y_4	$\boldsymbol{x_2}$	x_3	$\boldsymbol{x_4}$	x_5	$\boldsymbol{x_0}$	x_1	x_2	$\boldsymbol{x_3}$	x_4	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_1}$
y_5	$\boldsymbol{x_1}$	x_2	$\boldsymbol{x_3}$	x_4	$\boldsymbol{x_5}$	x_0	x_1	$\boldsymbol{x_2}$	x_3	$\boldsymbol{x_4}$	x_5	$\boldsymbol{x_0}$

Table 4: Extending the left super-game from $U_i = \{\{x_0, x_1, x_2\}, \{x_3, x_4, x_5\}\}$ to $U_j = \{\{y_0, y_1, y_2\}, \{y_3, y_4, y_5\}\}$ into normal games in $2kd = 12$ days, where home normal games are marked in bold

normal teams in U_i and U_j, we define matches

$$s_i = \{x_{i'} \to y_{(kd+i-i') \bmod kd}\}_{i'=0}^{kd-1}.$$

We use $\overline{s_i}$ to denote the case that the game venues in s_i are reversed. We directly extend the left super-game into normal games between normal teams in U_i and U_j. The normal games can be presented by

$$s_0\overline{s_1}s_2\overline{s_3}\cdots s_{kd-2}\overline{s_{kd-1}} \cdot \overline{s_0\overline{s_1}s_2\overline{s_3}\cdots s_{kd-2}\overline{s_{kd-1}}}.$$

An illustration of the normal games after extending one left super-game for $k = 3$ and $d = 2$ is shown in Table 4. Note that all normal games between one normal team in U_i and one in U_j are arranged.

Right super-games: Consider that super-team U_i plays an away right super-game with the super-team U_j together with a team-pair $R_{i'}$ (the direction of the edge is from U_i to U_j). Let $U_i = \{\{x_0, \dots, x_{k-1}\}, \dots, \{x_{kd-k}, \dots, x_{kd-1}\}\}$, $U_j = \{\{y_0, \dots, y_{k-1}\}, \dots, \{y_{kd-k}, \dots, y_{kd-1}\}\}$, and $R_{i'} = \{t_{n_\varepsilon+2i'-1}, t_{n_\varepsilon+2i'}\}$. We will put the first team $t_{n_\varepsilon+2i'-1} \in R_{i'}$ to the upper super-team as shown in Figure 1 or 2 and put the second team $t_{n_\varepsilon+2i'} \in R_{i'}$ to the

	1	2	3	4	5	6	7	8	9	10	11	12
x_0	y_0	$\boldsymbol{y_2}$	y_4	$\boldsymbol{y_6}$	y_1	$\boldsymbol{y_3}$	y_2	$\boldsymbol{y_4}$	y_6	$\boldsymbol{y_1}$	y_3	$\boldsymbol{y_0}$
x_1	y_6	$\boldsymbol{y_1}$	y_3	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_2}$	y_1	$\boldsymbol{y_3}$	y_5	$\boldsymbol{y_0}$	y_2	$\boldsymbol{y_6}$
x_2	y_5	$\boldsymbol{y_0}$	y_2	$\boldsymbol{y_4}$	y_6	$\boldsymbol{y_1}$	y_0	$\boldsymbol{y_2}$	y_4	$\boldsymbol{y_6}$	y_1	$\boldsymbol{y_5}$
x_3	y_4	$\boldsymbol{y_6}$	y_1	$\boldsymbol{y_3}$	y_5	$\boldsymbol{y_0}$	y_6	$\boldsymbol{y_1}$	y_3	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_4}$
x_4	y_3	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_2}$	y_4	$\boldsymbol{y_6}$	y_5	$\boldsymbol{y_0}$	y_2	$\boldsymbol{y_4}$	y_6	$\boldsymbol{y_3}$
x_5	y_2	$\boldsymbol{y_4}$	y_6	$\boldsymbol{y_1}$	y_3	$\boldsymbol{y_5}$	y_4	$\boldsymbol{y_6}$	y_1	$\boldsymbol{y_3}$	y_5	$\boldsymbol{y_2}$
x_6	y_1	$\boldsymbol{y_3}$	y_5	$\boldsymbol{y_0}$	y_2	$\boldsymbol{y_4}$	y_3	$\boldsymbol{y_5}$	y_0	$\boldsymbol{y_2}$	y_4	$\boldsymbol{y_1}$
y_0	$\boldsymbol{x_0}$	x_2	$\boldsymbol{x_4}$	x_6	$\boldsymbol{x_1}$	x_3	$\boldsymbol{x_2}$	x_4	$\boldsymbol{x_6}$	x_1	$\boldsymbol{x_3}$	x_0
y_1	$\boldsymbol{x_6}$	x_1	$\boldsymbol{x_3}$	x_5	$\boldsymbol{x_0}$	x_2	$\boldsymbol{x_1}$	x_3	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_2}$	x_6
y_2	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_2}$	x_4	$\boldsymbol{x_6}$	x_1	$\boldsymbol{x_0}$	x_2	$\boldsymbol{x_4}$	x_6	$\boldsymbol{x_1}$	x_5
y_3	$\boldsymbol{x_4}$	x_6	$\boldsymbol{x_1}$	x_3	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_6}$	x_1	$\boldsymbol{x_3}$	x_5	$\boldsymbol{x_0}$	x_4
y_4	$\boldsymbol{x_3}$	x_5	$\boldsymbol{x_0}$	x_2	$\boldsymbol{x_4}$	x_6	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_2}$	x_4	$\boldsymbol{x_6}$	x_3
y_5	$\boldsymbol{x_2}$	x_4	$\boldsymbol{x_6}$	x_1	$\boldsymbol{x_3}$	x_5	$\boldsymbol{x_4}$	x_6	$\boldsymbol{x_1}$	x_3	$\boldsymbol{x_5}$	x_2
y_6	$\boldsymbol{x_1}$	x_3	$\boldsymbol{x_5}$	x_0	$\boldsymbol{x_2}$	x_4	$\boldsymbol{x_3}$	x_5	$\boldsymbol{x_0}$	x_2	$\boldsymbol{x_4}$	x_1

Table 5: Extending the right super-game from $U_i = \{\{x_0, x_1, x_2\}, \{x_3, x_4, x_5\}\}$ to $U_j = \{\{y_0, y_1, y_2\}, \{y_3, y_4, y_5\}\}$ into normal games in $2kd = 12$ days, where home normal games are marked in bold

lower super-team: if U_i is the upper super-team, then let $t_{n_\varepsilon+2i'-1} = x_{kd}$ and $t_{n_\varepsilon+2i'} = y_{kd}$, and otherwise, let $t_{n_\varepsilon+2i'} = x_{kd}$ and $t_{n_\varepsilon+2i'-1} = y_{kd}$. Recall that kd is divisible by 2 since d is an even constant. For all normal teams in U_i, U_j and $R_{i'}$, we define matches

$$s_i = \{x_{i'} \to y_{(kd+1+2i-i') \bmod (kd+1)}\}_{i'=0}^{kd}.$$

The extended normal games can be presented by

$$s_0\overline{s_1}s_2\overline{s_3}\cdots s_{kd-2}\overline{s_{kd-1}} \cdot \overline{\overline{s_1}s_2\overline{s_3}s_4\cdots\overline{s_{kd-1}}s_0}.$$

An illustration of the normal games after extending one right super-game for $k = 3$ and $d = 2$ is shown in Table 5.

For each right super-game, we have two more normal teams, and then two days of normal games may not be able to arrange in the super-game. In fact, we did not arrange the matches s_{kd} and $\overline{s_{kd}}$, which contain the two games between x_{kd} and y_{kd}. They will be arranged in the last time slot.

The Last Time Slot

Next, we consider the schedule in the last time slot. Recall that the last time slot contains $2n-2-2kd(m-2)$ days. We also split the last time slot into two sessions. The first session contains $2kd$ days, which hold $m/2$ super-games similar to these in the first $m-2$ time slots. The second session will schedule all unarranged games.

For the first session, we have $m/2 - r$ left super-games (all previous middle super-games become left super-games) and r right super-games, as shown in Figure 3. We denote the games in the first session by

$$\Upsilon = \Upsilon_1 \cdot \Upsilon_2 \cdots \Upsilon_{kd},$$

where Υ_i represents the matches on days $2i-1$ and $2i$. We will use 'A' (resp., 'H') to indicate the state of one team playing an away (resp., a home) game. According to the design of left and right super-games (see Tables 4 and 5), we know that the states of each team in Υ_i are AH or HA.

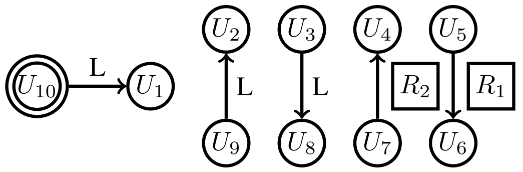


Figure 3: The super-game schedule in the last time slot, where $m = 10$ and $r = 2$

Next, we consider all unarranged games in the second session.

The unarranged games: Part-1. We consider the games involving teams in the last super-team U_m and the r team-pairs. According to the previous construction, we know that no pair of teams in this part has played a game. Thus, we only need to arrange a double round-robin for them. There are $kd+2r = n-(m-1)kd$ teams in total. Hence, the games will span $2n-2(m-1)kd-2$ days. We can simply call an algorithm of TTP-2 (for example, the algorithm in (Thielen and Westphal 2012)) for them to arrange the games to satisfy the no-repeat and bounded-by-2 constraints. The matches of the $n-(m-1)kd$ teams on these $2n-2(m-1)kd-2$ days are denoted by Φ.

The unarranged games: Part-2. We consider the unarranged games involving teams in super-teams in $\{U_1, \ldots, U_{m-1}\}$. We further split games into two sub-parts: **Part 2.1** and **Part 2.2**.

Part 2.1 is the unarranged games within each super-team. For each super-team, no pair of teams in it has played a game and then we also arrange a double round-robin for the teams in each super-team. There are $m-1$ super-teams and each super-team contains kd teams. Similarly, for each super-team, we call an algorithm of TTP-2 to arrange the games satisfying the no-repeat and bounded-by-2 constraints. This will span $2kd-2$ days. The matches of the $(m-1)kd$ teams on these $2kd-2$ days are denoted by Ψ_1.

Part 2.2 is the unarranged games left in the right super-games. Note that when $r = 0$, **Part 2.2** does not exist. We simply assume $r > 0$. In each right super-game, only matches s_{kd} and $\overline{s_{kd}}$ were not arranged.

There are r right super-games, and each of them involves one team-pair. The right super-game involving team-pair R_i is called the i-th right super-game. Suppose there is a i-th right super-game between super-teams U_j and $U_{j'}$, then we color the super-edge $U_jU_{j'}$ with color i. Since each super-team of $\{U_1, \ldots, U_{m-1}\}$ plays exactly two i-th right super-games, we know the super-edges with color i form a set of disjoint cycles, denoted by $\mathcal{C}_i$.

For example, when $m = 10$ and $r = 2$ (see Figure 3), $\mathcal{C}_1$ contains one cycle $U_1U_2\ldots U_9$ and $\mathcal{C}_2$ contains three cycles $U_1U_4U_7$, $U_2U_5U_8$ and $U_3U_6U_9$. All these cycles are odd cycles, i.e., the cycle has an odd number of super-teams.

Lemma 5 (*). *$\mathcal{C}_i$ is a set of disjoint odd cycles on super-teams $\{U_1, \ldots, U_{m-1}\}$.*

We are ready to arrange the matches in **Part 2.2**. According to the r sets of cycles, the matches will be divided into r parts, denoted by $\Psi_2, \ldots, \Psi_{r+1}$.

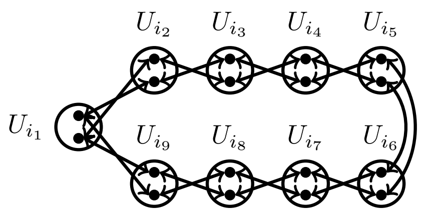


Figure 4: An illustration of the unarranged matches on the cycle C, where $p = 9$ and we only consider the first and the last team of each super-team

Without loss of generality, we consider the matches Ψ_{i+1} which is related to the cycle set $\mathcal{C}_i$. By Lemma 5, we know $\mathcal{C}_i$ is a set of odd cycles which also contains all of the $m-1$ super-teams $\{U_1, \ldots, U_{m-1}\}$. Let $C = U_{i_1}U_{i_2}\ldots U_{i_p}$ be an arbitrary odd cycle where p is odd. Due to the symmetric property of the unarranged matches s_{kd} and $\overline{s_{kd}}$, we only consider the first team and the last team in each super-team U_{i_j} of C. The unarranged matches on C are shown in Figure 4.

We can see that the matches form a bi-directed cycle with an even length on normal teams. It can be decomposed into four directed matchings s_1, s_2, $\overline{s_1}$ and $\overline{s_2}$, where $s_1 \cup s_2$ is a directed cycle and $\overline{s_1} \cup \overline{s_2}$ is the cycle with an opposite direction.

We can use four days to arrange the matches on C with $s_1 s_2 \overline{s_1}\,\overline{s_2}$. Note that the states of each team on these four days are either AHHA or HAAH. The games on four days over all cycles in $\mathcal{C}_i$ form the matches Ψ_{i+1}. Hence, the matches in **Part 2.2** can be presented by $\Psi_2 \cdot \Psi_3 \cdots \Psi_{r+1}$, which span $4r = 2n - 2mkd$ days. Note that there are at most two consecutive home/away games in **Part 2.2**.

The matches in **Part 2** can be presented by

$$\Psi = \Psi_1 \cdot \Psi_2 \cdots \Psi_{r+1}.$$

Recall that the matches in Ψ_1 span $2kd-2$ days. Hence, the matches in **Part 2** span $2kd-2+4r = 2n-2(m-1)kd-2$ days which is the same to Φ in **Part 1**.

According to Ψ_i, we can decompose $\Phi = \Phi_1 \cdot \Phi_2 \cdots \Phi_{r+1}$, where the days of Φ_i correspond to that of Ψ_i. If we further use Λ_i to denote the corresponding matches Φ_i and Ψ_i, the entire unarranged matches can be presented by

$$\Lambda = \Lambda_1 \cdot \Lambda_2 \cdots \Lambda_{r+1}.$$

Recall that the games in the first session of the last time slot are $\Upsilon = \Upsilon_1 \cdot \Upsilon_2 \cdots \Upsilon_{kd}$. The games in the second session are denoted by Λ. By putting them together, the games in the last time slot can be presented by

$$\Upsilon \cdot \Lambda = (\Upsilon_1 \cdot \Upsilon_2 \cdots \Upsilon_{kd}) \cdot (\Lambda_1 \cdot \Lambda_2 \cdots \Lambda_{r+1}).$$

Lemma 6 (*). *The games in the last time slot satisfy the bounded-by-3 property.*

Theorem 7 (*). *When $n \geq 2k^2d^2$, the above construction generates a feasible solution for TTP-k with any $k \geq 3$.*

The Analysis

Next, we analyze the total weight of our schedule and the approximation ratio. We first define some notations.

Recall that the k-path packing $\mathcal{P}^*_\varepsilon$ of G_ε contains md k-paths and each k-path P_i corresponds to a path-team u_i. Taking each path-team as a vertex, we define a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{u_1, \ldots, u_{kd}\}$. The weight of an edge u_iu_j ($i \neq j$) is defined to be the total weight of the edges in G_ε between one team in u_i and one team in u_j, i.e., $w(u_i, u_j) = \sum_{t_{i'} \in u_i, t_{j'} \in u_j} w(t_{i'}, t_{j'})$. We also define $w(u_i, u_i) = 0$ and $w(u_i) = \sum_{t_{i'}, t_{j'} \in u_i} w(t_{i'}, t_{j'})$. Recall that Δ_ε (resp., Δ) is twice the total weight of the edges in G_ε (resp., G). So,

$$w(\mathcal{E}) = \Delta_\varepsilon/2 - w(\mathcal{V}) \leq \Delta_\varepsilon/2 \leq \Delta/2. \qquad (2)$$

Since the labels of path-teams are obtained randomly, by (2), we have

$$\mathbb{E}[w(u_i, u_j)] = \frac{1}{|\mathcal{E}|} w(\mathcal{E}) \leq \frac{2}{m^2d^2}\Delta_\varepsilon, \qquad (3)$$

where it follows from that $|\mathcal{E}| = \frac{|\mathcal{V}|(|\mathcal{V}|-1)}{2} = \frac{md(md-1)}{2} \geq \frac{m^2d^2}{4}$ since $md \geq 2kd^2 \geq 24$ (recall that $k \geq 3$, $d \geq 2$, and $m \geq 2kd$).

Taking each super-team as a vertex, we can define a similar graph $\mathcal{H} = (\mathcal{U}, \mathcal{F})$. The weight of an edge U_iU_j ($i \neq j$) is defined as $w(U_i, U_j) = \sum_{u_{i'} \in U_i, u_{j'} \in U_j} w(u_{i'}, u_{j'})$. Define $w(U_i, U_i) = 0$ and $w(U_i)$ be the total weight of all edges between any pair of vertices in $\bigcup_{u_{i'} \in U_i} u_{i'}$. We can get that $\mathbb{E}[w(U_i, U_j)] = \sum_{u_{i'} \in U_i, u_{j'} \in U_j} \mathbb{E}[w(u_{i'}, u_{j'})] = d^2 \cdot \mathbb{E}[w(u_{i'}, u_{j'})]$. By (3), we have

$$\mathbb{E}[w(U_i, U_j)] \leq (2/m^2)\Delta_\varepsilon. \qquad (4)$$

For the sake of analysis, we assume that all teams return home before and after each day's game in left super-games, right super-games, and the last time slot. For example, assuming that there are two games between teams t_i and t_j in the last time slot, the weight of them will be $4w(t_i, t_j)$ after the assumption. In middle super-games, we assume that all teams of it return home before the first day and after the last day. By the triangle inequality, these assumptions will not decrease the weight of our schedule.

We will consider the weight of the following four parts:

- W_a: the weight in games involving teams in team-pairs;
- W_b: the weight in games within super-teams;
- W_c: the weight in games in left and right super-games;
- W_d: the weight in games in middle super-games.

First, we consider W_a. The games are arranged in the right super-games and **Part 1**. There are $2r$ teams in r team-pairs. For each team t_i of it, every other team $t_j \in V \setminus \{t_i\}$ plays one away game and one home game with it in right super-games or the last time slot. By the assumption, t_i and t_j return home before and after each of these two games. The weight of them is $4w(t_i, t_j)$. Hence, the weight of games related to t_i is exactly $4\delta(t_i)$. The weight W_a is bounded by $\sum_{t_i \in V \setminus V_\varepsilon} 4\delta(t_i)$. Note that $\sum_{t_i \in V_\varepsilon} \delta(t_i) \geq (mkd/n)\Delta$

by (1), $\sum_{t_i \in V} \delta(t_i) = \Delta$, and $2r = n - n_\varepsilon = n - mkd$ by definitions. Hence, we can get that $\sum_{t_i \in V \setminus V_\varepsilon} 4\delta(t_i) \leq 4(1 - mkd/n)\Delta = (8r/n)\Delta$. Recall that $r = kd - 1 \leq kd$ and $\Delta \leq (k/2) \cdot \text{OPT}$ by Lemma 1. We have

$$\mathbb{E}[W_a] \leq (8r/n)\Delta = (4k^2d/n) \cdot \text{OPT}. \tag{5}$$

Second, we consider W_b. The games within super-teams are arranged in **Part 1** and **Part 2.1**. For each super-team U_i, by the assumption, the weight of games is $4w(U_i)$. Hence, $W_b = 4\sum_{i=1}^{m} w(U_i)$. By the triangle inequality, we can get that $w(U_i) + w(U_j) \leq 2w(U_i, U_j)$ for any $i \neq j$. By (4), we have that $\mathbb{E}[w(U_i) + w(U_j)] \leq (4/m^2)\Delta_\varepsilon$ and hence $\sum_{i=1}^{m} 4\mathbb{E}[w(U_i)] \leq (8/m)\Delta_\varepsilon \leq (8/m)\Delta$. Recall that $m = 2\lfloor \frac{n}{2kd} \rfloor \geq \frac{n}{kd} - 2 \geq \frac{n}{2kd}$ since $n \geq 2k^2d^2 \geq 4kd$ ($k \geq 3$ and $d \geq 2$). We have

$$\mathbb{E}[W_b] \leq (8/m)\Delta \leq (8k^2d/n) \cdot \text{OPT}. \tag{6}$$

Third, we consider W_c. Recall that $r \leq kd - 1 \leq kd$. Hence, there are $(m-2) + (m/2 - r) \leq 2m \leq mkd$ left and $(m-1)r \leq mr \leq mkd$ right super-games. Note that we take the games in **Part 2.2** as a part of right super-games. In each left/right super-game between super-teams U_i and U_j, by the assumption, the weight of all games is $4w(U_i, U_j)$. By (4), we have $4\mathbb{E}[w(U_i, U_j)] = (8/m^2)\Delta_\varepsilon$. Hence, the expected weight of mkd left and mkd right super-games is $(16kd/m)\Delta_\varepsilon$. Recall that $m \geq \frac{n}{2kd}$. We have

$$\mathbb{E}[W_c] \leq (16kd/m)\Delta \leq (16k^3d^2/n) \cdot \text{OPT}. \tag{7}$$

At last, we consider W_d. For the sake of analysis, we assume each super-team plays $m-1$ middle super-games with the other $m-1$ super-teams, which can only increase the total weight of our schedule.

By Lemma 4, if every team in G_ε plays k-consecutive away games along each k-path of $\mathcal{P}_\varepsilon^*$, then the total traveling distance of all teams is exactly $\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) + n_\varepsilon w(\mathcal{P}_\varepsilon^*) = \text{OPT}$. We call such a schedule an *ideal schedule*. Our construction is similar to the ideal schedule and hence we only need to calculate the weight of different parts.

Consider each middle super-game separately. Recall that we assume all teams return home before the first day and after the last day in the super-game. By the design of middle super-games, every team plays d or $d+1$ away trips which follows at least $d-1$ k-paths. There are at most $m-1$ middle super-games for it and md k-paths in $\mathcal{P}_\varepsilon^*$. Hence, for an arbitrary team $x_{i'}$ of path-team u_i, it does not follow at most m k-paths. Suppose it does not follow the k-path of path-team $u_j = \{y_0, \ldots, y_{k-1}\}$. Then it plays two away trips on the k-path, and we assume that the two corresponding segments are $\{y_0, \ldots, y_{k'-1}\}$ and $\{y_{k'}, \ldots, y_{k-1}\}$. Note that these two segments follow from two sub-parts of the k-path (see Figure 3). Comparing with the trip on $\{y_0, \ldots, y_{k-1}\}$ in the ideal schedule, the weight of the different parts is $w(x_{i'}, y_{k'-1}) + w(x_{i'}, y_{k'}) - w(y_{k'-1}, y_{k'}) \leq w(x_{i'}, y_{k'-1}) + w(x_{i'}, y_{k'}) \leq (2/k^2)w(u_i, u_j) + 2(w(P_i) + w(P_j))$ by the triangle inequality. There are md k-paths in $\mathcal{P}_\varepsilon^*$. We have $2\mathbb{E}[w(P_i)] = 2\mathbb{E}[w(P_j)] = \frac{2}{md}w(\mathcal{P}_\varepsilon^*)$. By (3), we have $\frac{2}{k^2}\mathbb{E}[w(u_i, u_j)] \leq \frac{4}{m^2d^2k^2}\Delta_\varepsilon = \frac{4}{n_\varepsilon^2}\Delta_\varepsilon$. Hence, the expected weight of the different parts is bounded by $\frac{4}{n_\varepsilon^2}\Delta_\varepsilon + \frac{4}{md}\mathcal{P}_\varepsilon^*$. Since it does not follow at most m k-paths, for team $x_{i'}$, the total expected weight of the different parts is at most $\frac{4m}{n_\varepsilon^2}\Delta_\varepsilon + \frac{4}{d}\mathcal{P}_\varepsilon^* = \frac{4}{n_\varepsilon kd}\Delta_\varepsilon + \frac{4}{d}\mathcal{P}_\varepsilon^*$. For all n_ε teams of G_ε, the total expected weight of the different parts is at most $\frac{4}{kd}\Delta_\varepsilon + \frac{4}{d}n_\varepsilon\mathcal{P}_\varepsilon^*$. Note that $\frac{4}{kd}\Delta_\varepsilon \leq \frac{4}{kd}\Delta \leq \frac{2}{d}\cdot\text{OPT}$. By Lemma 4, we can get that $\frac{4}{d}n_\varepsilon\mathcal{P}_\varepsilon^* \leq \frac{4}{d}(\Delta_\varepsilon(\mathcal{P}_\varepsilon^*) + n_\varepsilon\mathcal{P}_\varepsilon^*) \leq \frac{4}{d} \cdot \text{OPT}$. We have

$$\mathbb{E}[W_d] \leq (1 + 6/d) \cdot \text{OPT}. \tag{8}$$

Theorem 8. *For any $3 \leq k = o(\sqrt[3]{n})$, LDTTP-$k$ allows an EPTAS: for any constant $\varepsilon > 0$, we can compute a schedule for LDTTP-k with the weight at most $1 + \varepsilon$ times the optimal in time $O_\varepsilon(1) + n^{O(1)}$.*

Proof. First, our schedule takes $O(n^2)$ time to find the vertex set V_ε and complete the construction. Second, by (5), (6), (7), (8), $kd \geq 6$, and $12k^2d/n \leq 2k^3d^2/n$, we know the total weight is bounded by $(1 + 4k^2d/n + 8k^2d/n + 16k^3d^2/n + 6/d) \cdot \text{OPT} \leq (1 + 18k^3d^2/n + 6/d) \cdot \text{OPT}$. Recall that $d = 12\lceil 1/\varepsilon \rceil \geq 12/\varepsilon$, we have $(1 + 18k^3d^2/n + 6/d) \cdot \text{OPT} \leq (1 + 18k^3d^2/n + \varepsilon/2) \cdot \text{OPT}$. Furthermore, since $k = o(\sqrt[3]{n})$ and $d = O_\varepsilon(1)$, we know there is a constant $n_1 = O_\varepsilon(1)$ such that $18k^3d^2/n \leq \varepsilon/2$ when $n \geq n_1$. Recall the constant $n_0 = O_\varepsilon(1)$. When $n \leq \max\{n_0, n_1\}$, we can simply use a brute-and-force algorithm to find an optimal solution which takes constant time $O_\varepsilon(1)$. Otherwise, our schedule takes $O(n^2)$ time. The total weight is bounded by $(1 + \varepsilon/2 + \varepsilon/2) \cdot \text{OPT} = (1 + \varepsilon) \cdot \text{OPT}$ and the running time is bounded by $O_\varepsilon(1) + n^2$, which implies that it is an EPTAS.

Our algorithm is random since the path-teams are labeled randomly. It can be derandomized efficiently by the method of conditional expectations (Motwani and Raghavan 1995), which takes an additional $n^{O(1)}$ time. The derandomization is omitted. □

When k is a constant, we have $k = O(1)$ and then we can get that $n_0 = O(1/\varepsilon^2)$ and $n_1 = O(1/\varepsilon^3)$. By enumerating all $n(n-1)$ games on $2(n-1)$ days, a trivial brute-and-force algorithm takes $(2(n-1))^{n(n-1)} = 2^{O(n^2 \log n)}$ time. Therefore, the running time of the brute-and-force algorithm for $n \leq \max\{n_0, n_1\}$ is $2^{O(1/\varepsilon^6 \log(1/\varepsilon))}$ and the running time of the randomized algorithm is bounded by $2^{O(1/\varepsilon^6 \log(1/\varepsilon))} + O(n^2)$.

Conclusion

In this paper, we propose a novel construction for TTP-k. For LDTTP-k with any $3 \leq k = o(\sqrt[3]{n})$, it generates an EPTAS, greatly improving previous known approximation algorithms for LDTTP-k. The paper focuses on theoretical analysis. It is also worth studying the experimental performance of our construction. We mark that the larger the number n of teams, the better performance of our schedule. For small n, our algorithm suggests the brute-and-force method, which will not be practical.

Acknowledgments

The work is supported by the National Natural Science Foundation of China, under grants 61972070.

References

Anagnostopoulos, A.; Michel, L.; Van Hentenryck, P.; and Vergados, Y. 2006. A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2): 177–193.

Bhattacharyya, R. 2016. Complexity of the unconstrained traveling tournament problem. *Operations Research Letters*, 44(5): 649–654.

Bulck, D. V.; Goossens, D. R.; Schönberger, J.; and Guajardo, M. 2020. RobinX: A three-field classification and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, 280(2): 568–580.

Chatterjee, D. 2021. Complexity of Traveling Tournament Problem with Trip Length More Than Three. arXiv:2110.02300.

Chatterjee, D.; and Roy, B. K. 2021. An Improved Scheduling Algorithm for Traveling Tournament Problem with Maximum Trip Length Two. In *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2021*, volume 96 of *OASIcs*, 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Di Gaspero, L.; and Schaerf, A. 2007. A composite-neighborhood tabu search approach to the traveling tournament problem. *Journal of Heuristics*, 13(2): 189–207.

Easton, K.; Nemhauser, G. L.; and Trick, M. A. 2001. The Traveling Tournament Problem Description and Benchmarks. In *Principles and Practice of Constraint Programming, 7th International Conference, CP 2001, Proceedings*, volume 2239 of *Lecture Notes in Computer Science*, 580–584. Springer.

Easton, K.; Nemhauser, G. L.; and Trick, M. A. 2002. Solving the Travelling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach. In *Practice and Theory of Automated Timetabling IV, 4th International Conference, PATAT 2002*, volume 2740 of *Lecture Notes in Computer Science*, 100–112. Springer.

Goerigk, M.; Hoshino, R.; Kawarabayashi, K.; and Westphal, S. 2014. Solving the Traveling Tournament Problem by Packing Three-Vertex Paths. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2271–2277. AAAI Press.

Goerigk, M.; and Westphal, S. 2016. A combined local search and integer programming approach to the traveling tournament problem. *Annals of Operations Research*, 239(1): 343–354.

Hentenryck, P. V.; and Vergados, Y. 2007. Population-Based Simulated Annealing for Traveling Tournaments. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 267–272. AAAI Press.

Hoshino, R.; and Kawarabayashi, K. 2012. The Linear Distance Traveling Tournament Problem. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 1770–1778. AAAI Press.

Hoshino, R.; and Kawarabayashi, K. 2013. An Approximation Algorithm for the Bipartite Traveling Tournament Problem. *Mathematics of Operations Research*, 38(4): 720–728.

Imahori, S. 2021. A 1+O(1/N) approximation algorithm for TTP(2). arXiv:2108.08444.

Imahori, S.; Matsui, T.; and Miyashiro, R. 2014. A 2.75-approximation algorithm for the unconstrained traveling tournament problem. *Annals of Operations Research*, 218(1): 237–247.

Kendall, G.; Knust, S.; Ribeiro, C. C.; and Urrutia, S. 2010. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, 37(1): 1–19.

Lim, A.; Rodrigues, B.; and Zhang, X. 2006. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3): 1459–1478.

Miyashiro, R.; Matsui, T.; and Imahori, S. 2012. An approximation algorithm for the traveling tournament problem. *Annals of Operations Research*, 194(1): 317–324.

Motwani, R.; and Raghavan, P. 1995. *Randomized algorithms*. Cambridge university press.

Thielen, C.; and Westphal, S. 2011. Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4): 345–351.

Thielen, C.; and Westphal, S. 2012. Approximation algorithms for TTP(2). *Mathematical Methods of Operations Research*, 76(1): 1–20.

Trick, M. 2020. Challenge traveling tournament instances. http://mat.gsia.cmu.edu/TOURN/. Accessed: 2022-8-15.

Westphal, S.; and Noparlik, K. 2014. A 5.875-approximation for the traveling tournament problem. *Annals of Operations Research*, 218(1): 347–360.

Xiao, M.; and Kou, S. 2016. An Improved Approximation Algorithm for the Traveling Tournament Problem with Maximum Trip Length Two. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016*, volume 58 of *LIPIcs*, 89:1–89:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Yamaguchi, D.; Imahori, S.; Miyashiro, R.; and Matsui, T. 2011. An improved approximation algorithm for the traveling tournament problem. *Algorithmica*, 61(4): 1077–1091.

Zhao, J.; and Xiao, M. 2021a. A Further Improvement on Approximating TTP-2. In *Computing and Combinatorics - 27th International Conference, COCOON 2021, Proceedings*, volume 13025 of *Lecture Notes in Computer Science*, 137–149. Springer.

Zhao, J.; and Xiao, M. 2021b. The Traveling Tournament Problem with Maximum Tour Length Two: A Practical Algorithm with An Improved Approximation Bound. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, 4206–4212.

Zhao, J.; Xiao, M.; and Xu, C. 2022. Improved Approximation Algorithms for the Traveling Tournament Problem. In *MFCS 2022*, volume 241 of *LIPIcs*, 83:1–83:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.