A TSP-Based Algorithm for Multi-League Traveling Tournament

Jingyang Zhao[1,2], **Mingyu Xiao**[1*], **Ken-ichi Kawarabayashi**[3,4]
[1]University of Electronic Science and Technology of China
[2]Kyung Hee University, Yongin-si, South Korea
[3]National Institute of Informatics
[4]The University of Tokyo, Tokyo, Japan
jingyangzhao1020@gmail.com, myxiao@uestc.edu.cn, k_keniti@nii.ac.jp

Abstract

In some professional sports leagues, inter-league games are scheduled among multiple divisions or conferences. This inspired us to study the p-partite Traveling Tournament Problem (p-partite TTP), where all teams are partitioned into p leagues, and each team plays games against all teams from other leagues. Previously, only the case of $p = 2$, known as the Bipartite TTP or BTTP, has been introduced and studied. In this paper, we show that the p-partite TTP is NP-hard for any fixed $p \geq 3$, and we propose an efficient algorithm based on a solution to the Traveling Salesman Problem. Furthermore, we prove that the algorithm achieves a notable approximation ratio of $\frac{8}{3} + O(\frac{1}{n})$ when $p = 3$. We also conduct experiments demonstrating that the algorithm produces practical schedules with significantly reduced total travel distances, highlighting its effectiveness in generating high-quality multipartite tournament schedules.

Code — https://github.com/JingyangZhao/PTTP

Introduction

In sports scheduling, where teams must travel long distances to play their games, designing schedules that minimize total travel distance is crucial for both economic and environmental reasons. In this context, the Traveling Tournament Problem (TTP), introduced by the head schedulers of Major League Baseball (MLB) (Easton, Nemhauser, and Trick 2001), has emerged as a prominent scheduling problem and has inspired extensive research in Artificial Intelligence and Operations Research over the past three decades (Kendall et al. 2010; Durán 2021).

The TTP addresses the trade-off between game scheduling and travel minimization, aiming to construct a feasible double round-robin (DRR) tournament with minimized total travel distance. It has practical applications in professional sports leagues such as MLB and Nippon Professional Baseball (NPB), where the optimized schedules can reduce costs, save time, and lower the greenhouse gas emissions (Hoshino and Kawarabayashi 2012a,b).

Given a set of n (even) teams, a DRR tournament spans $2(n-1)$ days, where each team plays one game per day and plays against every other team twice (once at home and once away). The objective of TTP is to determine a DRR tournament that minimizes the total travel cost incurred by all teams, subject to the following constraints:

1. The *no-repeat* constraint: No team pair plays consecutive games against one another;
2. The *at-most-three* constraint: Each team plays at most three consecutive home or away games.

The TTP was proven to be NP-hard by Thielen and Westphal (2011) and even APX-hard by Zhao and Xiao (2025a). It is a highly challenging optimization problem, as establishing the optimality of solutions is significantly difficult. For example, there is still an open instance with $n = 12$ teams on the online benchmark website (Trick 2025; Bulck et al. 2020). Currently, the TTP has been tackled by numerous approaches, including Lagrangian relaxation (Benoist, Laburthe, and Rottembourg 2001), constraint and integer programming (Easton, Nemhauser, and Trick 2002; Goerigk and Westphal 2016), simulated annealing (Lim, Rodrigues, and Zhang 2006; Hentenryck and Vergados 2007), beam search (Frohner et al. 2023), tabu search (Gaspero and Schaerf 2007), approximation algorithms (Westphal and Noparlik 2014; Zhao and Xiao 2025c,b), and hybrid algorithms (Goerigk et al. 2014).

Inspired by the widespread use of a two-conference structure in many professional sports leagues, such as the "Big Four" leagues of North America: MLB, the National Basketball Association (NBA), the National Football League (NFL), and the National Hockey League (NHL), Hoshino and Kawarabayashi (2011a; 2011b) introduced an interesting *inter-league extension* of the TTP, called the Bipartite Traveling Tournament Problem (BTTP). In this problem, $2n$ teams are partitioned into two leagues of n teams each, and the objective is to construct a "distance-optimal" DRR *bipartite tournament* over $2n$ days. In such a tournament, every team in one league plays both a home and an away game against each team in the other league, while satisfying the no-repeat and at-most-three constraints.

Although many professional sports leagues adopt a two-conference structure, some use three or more conferences or divisions. For example, Super Rugby has historically organized teams into three conferences (Australia, New Zealand, and South Africa). Moreover, based on geography, the 30

---

[*] Corresponding author

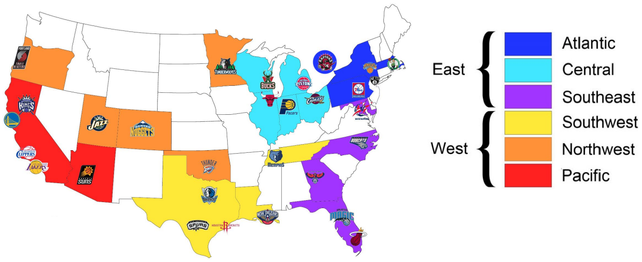


Figure 1: The six divisions of the NBA, where each division consists of 5 teams.

teams in the NBA are divided into six divisions of 5 teams each, and the 32 teams in the NFL are divided into eight divisions of 4 teams each. An illustration of the (previous) six divisions of the NBA can be found in Fig. 1.

Motivated by these observations, we consider the more general p-partite TTP, where pn teams are partitioned into p leagues of n teams each. The aim is to find a distance-optimal DRR *p-partite tournament* over $2(p-1)n$ days, in which every team in one league plays one home game and one away game against each team in all other leagues and satisfies the no-repeat and at-most-three constraints. The case that $p=2$ becomes BTTP.

Similar to BTTP, to compute the total travel distance, we assume that each team is at its home venue before the first game starts and returns home after the last game ends. On consecutive days, each team travels directly from the venue of its first game to that of its next game. For any two teams x and y, let $c(x,y)$ denote the distance between their home venues, where c is assumed to be a metric.

Example 1. *For instance, Table 1 shows a solution to BTTP with two leagues $\{x_0, x_1\}$ and $\{y_0, y_1\}$, where x_0 plays two consecutive away games on the first two days, followed by two consecutive home games on the last two days. The total travel cost of x_0 is $c(x_0, y_0) + c(y_0, y_1) + c(y_1, x_0)$.*

	1	2	3	4
x_0	y_0	y_1	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$
x_1	y_1	y_0	$\boldsymbol{y_1}$	$\boldsymbol{y_0}$
y_0	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	x_0	x_1
y_1	$\boldsymbol{x_1}$	$\boldsymbol{x_0}$	x_1	x_0

Table 1: A solution to BTTP with $p=2$ leagues $\{x_0, x_1\}$ and $\{y_0, y_1\}$, where home games are marked in bold.

It is worth noting that some other studies on multi-league sports scheduling can be found in (Li, Davari, and Goossens 2023; Li, Bulck, and Goossens 2025; Li and Goossens 2025). However, these works primarily focus on partitioning teams into leagues and then constructing schedules to minimize the overall cost, while considering competitive balance constraints and other practical factors, which differ from the p-partite TTP studied in this paper.

Our Contributions

In this paper, we initiate the study of the p-partite Traveling Tournament Problem (p-partite TTP) and prove that it is NP-hard for any fixed $p \geq 3$.

We then focus on designing effective algorithms for the p-partite TTP. Constructing even a simple feasible schedule for the p-partite TTP is non-trivial. To address this, we propose an approach based on the Traveling Salesman Problem (TSP) that applies to any fixed $p \geq 2$. While TSP-based methods have been used for the standard TTP in prior work (Yamaguchi et al. 2011; Westphal and Noparlik 2014), extending them to the p-partite TTP, even for $p=3$, is far from trivial. Furthermore, through a novel analysis, we show that our algorithm achieves an approximation ratio of $\frac{8}{3} + O(\frac{1}{n})$ for the 3-partite TTP. Finally, by incorporating several heuristic enhancements, we demonstrate that our algorithm delivers excellent empirical performance, producing high-quality tournament schedules in practice.

We emphasize that the 3-partite TTP is particularly relevant, as small values of p commonly arise in real-world applications, and the special case of $p=2$ (BTTP) has already been studied extensively.

Due to limited space, the proofs of lemmas marked with "*" were omitted, and they can be found in the full version of this paper.

Other Related Work

For TTP, BTTP, and their variants, there exists a lot of research on approximation algorithms in the literature (Thielen and Westphal 2012; Xiao and Kou 2016; Chatterjee and Roy 2021). Approximation approaches not only provide guaranteed solution quality but can also be combined with heuristic methods to produce high-quality practical solutions (Thielen and Westphal 2012; Westphal and Noparlik 2014; Goerigk et al. 2014; Goerigk and Westphal 2016).

For TTP, Miyashiro, Matsui, and Imahori (2012) developed the first $(2+O(\frac{1}{n}))$-approximation algorithm. This was improved to $\frac{5}{3} + O(\frac{1}{n})$ by Yamaguchi et al. (2011) and later to $\frac{139}{87} + \varepsilon$ by Zhao, Xiao, and Xu (2022). For BTTP, Hoshino and Kawarabayashi (2013) proposed the first $(2 + O(\frac{1}{n}))$-approximation algorithm, which was recently improved to $\frac{3}{2} + \varepsilon$ by Zhao and Xiao (2024).

Regarding TTP constraints, the at-most-three constraint is the most extensively studied setting and is adopted by well-known benchmark instances (Trick 2025; Bulck et al. 2020). Nonetheless, several works have investigated variants of TTP with different at-most-k constraints, including $k=2$ (Zhao and Xiao 2025c; Imahori 2021; Thielen and Westphal 2012), fixed $k \geq 4$ (Yamaguchi et al. 2011; Zhao and Xiao 2025b; Westphal and Noparlik 2014), and even $k=\infty$ (Imahori, Matsui, and Miyashiro 2014).

Preliminaries

We let $G=(V,E)$ denote the input complete graph, where V consists of np teams and is partitioned into p pairwise disjoint n-team leagues, denoted by $\mathcal{X} = \{X_1, ..., X_p\}$. Note that $\bigcup_{i=1}^{p} X_i = V$ and $X_i \cap X_j = \emptyset$ for all $1 \leq i < j \leq p$.

For any positive integer t, we let $[t] = \{1, 2, ..., t\}$. Then, for each $i \in [p]$, we define $\overline{X_i} = V \setminus X_i$ and $\overline{\mathcal{X}}_i = \mathcal{X} \setminus \{X_i\}$.

For any edge $xy \in E$, let $c(x, y)$ denote the distance between the home venues of teams x and y. We assume that the cost function c satisfies $c(x, x) = 0$, $c(x, y) = c(y, x)$ (symmetry) and $c(x, y) + c(y, z) \geq c(x, z)$ (triangle inequality) for all $x, y, z \in V$. Thus, c is a metric.

For any pair of disjoint sets $X, Y \subseteq V$, we let $c(X, Y) = \sum_{x\in X} \sum_{y\in Y} c(x, y)$. Moreover, if X consists only of a single vertex x, we let $c(x, Y) = c(X, Y)$. For any $X \subseteq V$, the complete graph induced by X is denoted by $G[X]$.

We use a sequence of distinct vertices to represent a *cycle*. For instance, the cycle $T = x_0x_1x_2...x_{t-1}$ consists of the t edges in $E(T) = \{x_{i-1}x_{i \bmod t} \mid i \in [t]\}$. The *size* of a cycle T is defined as $|E(T)|$, and its cost, denoted by $c(T)$, is the total cost of all edges in $E(T)$. In a graph $G' = (V', E')$, a *TSP cycle* is a cycle of size $|V'|$.

For two distinct teams x and y, we use $x \to y$ or $x \leftarrow y$ to denote a game between them at the home venue of x, and use $x \leftrightarrow y$ to denote the two games $x \to y$ and $x \leftarrow y$. As mentioned in the previous section, the goal of the p-partite TTP is to find a constrained distance-optimal DRR p-partite tournament (see Example 1). Therefore, in the p-partite TTP, we need to schedule all games in $\{x \leftrightarrow y \mid x \in X_i, y \in X_j, 1 \leq i < j \leq p\}$.

We fix an arbitrary optimal solution to the p-partite TTP. For any team $x \in V$, we let $\text{OPT}(x)$ denote its travel cost in this optimal solution. For any subset $X \subseteq V$, let $\text{OPT}(X) = \sum_{x\in X} \text{OPT}(x)$ and $\text{OPT} = \text{OPT}(V)$.

NP-Hardness and Approximability

The BTTP has been known to be NP-hard. It is thus unsurprising that the p-partite TTP is also NP-hard for any fixed $p \geq 2$, which we formally prove. The NP-hardness of BTTP was obtained via a complex reduction from 3-SAT (Hoshino and Kawarabayashi 2011a,b), but this approach does not easily generalize to the p-partite case. Instead, we adapt a reduction from 3-Tour Cover (3-TC) (Asano et al. 1997).

Definition 1 (3-TC). *An instance of 3-TC is represented by a complete graph $I = (V_I, E_I)$, a metric cost function c, and a parameter B, where V_I consists of a vertex (root) o and n other vertices (clients). The objective is to check whether there is a set of tours with cost at most B such that (1) each client is covered by only one tour, and (2) each tour is a cycle that includes the root and at most 3 clients.*

Note that 3-TC is also known as unit-demand Capacitated Vehicle Routing (Dantzig and Ramser 1959). We further impose the following two constraints on the input instance I, which do not affect its NP-hardness (Zhao and Xiao 2025a):

1. the function c outputs only *fixed* non-negative integers, i.e., $c : E_I \to \{0\} \cup [N]$ for some fixed integer $N > 0$;
2. n is divisible by 6, and there exists an optimal solution w.r.t. its minimization version, where each tour covers the root and exactly 3 clients.

Constraint 1 is natural, whereas Constraint 2 is enforced by adding dummy clients at the root's place to obtain an equivalent instance (see Lemma 8 in (Zhao and Xiao 2025a)).

Let $V'_I = V_I \setminus \{o\}$. Given an instance $I = (V_I, E_I)$ of 3-TC, we obtain an instance $J = (V_J, E_J)$ for the p-partite TTP, where $V_J = V'_I \cup \{o_1, ..., o_{pn^2-n}\}$ is obtained by adding the n clients in V'_I and $pn^2 - n$ copies of the root o. Hence, in J, there are pn^2 regular-teams, and the size of each league X_i ($i \in [p]$) is n^2. We assume $V'_I \subseteq X_1$. Clearly, the instance J can be constructed in polynomial time.

We assume $2c(o, V'_J) < n^2$; otherwise, since $c(o, V'_J) \leq nN$ by Constraint 1, we have $n \leq 2N$, i.e., n is a constant, and then I can be solved exactly in polynomial time.

Let $\text{OPT}(I)$ (resp., $\text{OPT}(J)$) be the optimal solution cost of the instance I (resp., J) w.r.t. its minimization version, and let $f(B) = (p-1)n^2B + 2(p-1)(\frac{n^2}{3} + 1)c(o, V'_I)$. To prove the NP-hardness of the p-partite TTP with fixed p, it suffices to prove the following lemma.

Lemma 1 (*). *$\text{OPT}(I) > B$ if and only if $\text{OPT}(J) > f(B)$.*

Theorem 1. *The p-partite TTP is NP-hard for any $p \geq 2$.*

While finding an optimal solution to the p-partite TTP is hard, any feasible solution achieves a 3-approximation. We prove this result via some simple observations and an extension of the classic TTP lower bound (Miyashiro, Matsui, and Imahori 2012).

Lemma 2. *For the p-partite TTP, any feasible solution has a cost at most 3 times the optimal cost.*

Proof. In any feasible solution, for each $i \in [p]$, each team $x \in X_i$ plays exactly one away game against every team in $\overline{X}_i$. By the triangle inequality, the total travel cost of x is at most $2\sum_{x'\in\overline{X}_i} c(x, x') = 2c(x, \overline{X}_i)$. Moreover, due to the at-most-three constraint, similar to TTP (Miyashiro, Matsui, and Imahori 2012; Westphal and Noparlik 2014), the travel cost of x is at least $\frac{2}{3}\sum_{x'\in\overline{X}_i} c(x, x') = \frac{2}{3}c(x, \overline{X}_i)$, which also implies that $\text{OPT}(x) \geq \frac{2}{3}c(x, \overline{X}_i)$. Hence, we obtain $2c(x, \overline{X}_i) \leq 3\text{OPT}(x)$. Summing over all teams in V, we know that any feasible solution has a cost at most 3OPT. ☐

Constructing a simple feasible solution is challenging, and breaking the 3-approximation barrier is non-trivial. We present a method to construct a feasible solution with an improved approximation ratio in the following sections.

The TSP-Based Construction

We present our construction approach based on the TSP technique for the TTP (Yamaguchi et al. 2011; Westphal and Noparlik 2014; Imahori, Matsui, and Miyashiro 2014). We first review this technique and then extend it to the p-partite TTP.

In TTP, a TSP-based algorithm computes an approximate TSP cycle in the graph G, aiming to schedule games so that nearly every team satisfies the *along-TSP* property:

- when a team plays consecutive away games, the opponents' locations appear consecutively in the TSP cycle.

Since the TSP cycle has a low cost, such algorithms ensure short travel distances and good approximations. For example, the algorithm in (Yamaguchi et al. 2011) achieves a $\frac{5}{3} + \varepsilon$ approximation, while the algorithm in (Westphal and

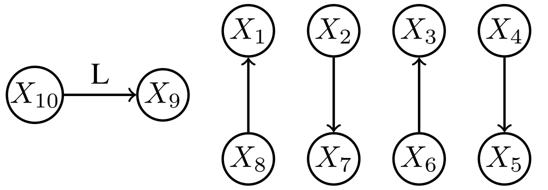


Figure 2: The SRR tournament in the 1st time slot ($p = 10$).

Noparlik 2014) improves benchmark solutions using heuristics.

In the p-partite TTP, each team in league X_i must play opponents from the other $p-1$ leagues in $\overline{X}_i$, making it difficult to compute a single TSP cycle in the graph G. Computing one in $G[\overline{X}_i]$ for each $i \in [p]$ yields p different cycles, complicating the task of ensuring that teams in X_i satisfy the along-TSP property relative to their respective cycles.

Therefore, our algorithm instead computes an approximate TSP cycle, denoted T_{X_i}, within each $G[X_i]$ for $i \in [p]$. The specifics are described in the next section; for now, we assume that the cycles T_{X_i} are given.

Finally, due to structural differences, our construction varies depending on whether p is even or odd. We first handle the even case, then extend the approach to the odd case with suitable modifications.

The Case where p is Even

Since the p-partite TTP reduces to the TTP when $n = 1$, we assume $n \geq 2$. Recall that there are p n-team leagues in $\mathcal{X}$. We refer to each league X_i as a *super-team*.

To avoid confusion, we will refer to each team in V as a *regular-team*, and a game between two regular-teams (resp., super-teams) as a *regular-game* (resp., *super-game*).

In our schedule construction, we first design a single round-robin (SRR) tournament for super-teams, where each distinct pair of super-teams plays exactly one super-game. Then, to obtain a feasible solution to the p-partite TTP, we extend super-games into regular-games.

Next, we first introduce the SRR tournament.

The SRR Tournament The SRR tournament construction is based on the classical *circle method* in sports scheduling (Yamaguchi et al. 2011; Imahori, Matsui, and Miyashiro 2014; Westphal and Noparlik 2014; Zhao and Xiao 2023; De Werra 1981). One key advantage of this method is that it may maximize the total number of consecutive home or away super-games for all participating super-teams.

Given p super-teams, the SRR tournament spans $p-1$ *time slots*. In each time slot, we schedule $p/2$ *super-games*, which include 1 *left* super-game and $p/2-1$ *normal* super-games. For ease of presentation, we use $X_i \rightarrow X_j$ (resp., $X_i \xrightarrow{\text{L}} X_j$) to denote a normal (resp., left) super-game between X_i and X_j, played at the home venue of X_j.

We now describe how these super-games are constructed.

In the 1st time slot, the $p/2$ super-games are scheduled as shown in Fig. 2, where the leftmost super-game is the left super-game, and the others are normal super-games.

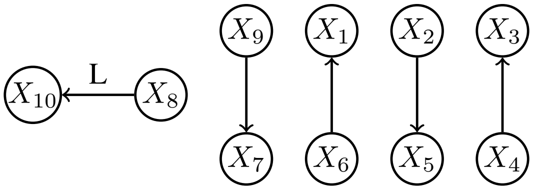


Figure 3: The SRR tournament in the 2nd time slot ($p = 10$).

	1	2	3	4	5	6	7	8	9
X_1	$\boldsymbol{X_8}$	$\boldsymbol{X_6}$	$\boldsymbol{X_4}$	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$	$\boldsymbol{X_5}$	$\boldsymbol{X_3}$	$\boldsymbol{X_{10}}$
X_2	X_7	X_5	X_3	X_1	X_8	X_6	X_4	X_{10}	$\boldsymbol{X_9}$
X_3	$\boldsymbol{X_6}$	$\boldsymbol{X_4}$	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$	$\boldsymbol{X_5}$	$\boldsymbol{X_{10}}$	X_1	X_8
X_4	X_5	X_3	X_1	X_8	X_6	X_{10}	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$
X_5	$\boldsymbol{X_4}$	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$	$\boldsymbol{X_{10}}$	X_3	X_1	X_8	X_6
X_6	X_3	X_1	X_8	X_{10}	$\boldsymbol{X_4}$	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$	$\boldsymbol{X_5}$
X_7	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_{10}}$	X_5	X_3	X_1	X_8	X_6	X_4
X_8	X_1	X_{10}	$\boldsymbol{X_6}$	$\boldsymbol{X_4}$	$\boldsymbol{X_2}$	$\boldsymbol{X_9}$	$\boldsymbol{X_7}$	$\boldsymbol{X_5}$	$\boldsymbol{X_3}$
X_9	$\boldsymbol{X_{10}}$	X_7	X_5	X_3	X_1	X_8	X_6	X_4	X_2
X_{10}	X_9	$\boldsymbol{X_8}$	X_7	$\boldsymbol{X_6}$	X_5	$\boldsymbol{X_4}$	X_3	$\boldsymbol{X_2}$	X_1

Table 2: An instance of the SRR tournament ($p = 10$), where home super-games are marked in bold.

In the 2nd time slot, the $p/2$ super-games are scheduled as shown in Fig. 3. We keep the position of the super-team X_p fixed, and cyclically shift the remaining super-teams in the cycle $X_1X_2...X_{p-1}$ by one position in the clockwise direction. Each edge is then reversed in direction. The super-games for the other time slots are constructed analogously.

An example of the super-games can be found in Fig. 2.

Next, we show how to extend the normal (and left) super-games into regular-games played between regular-teams.

The Extensions of Super-Games We first describe the extension of normal super-games.

Normal super-games: $X_i \rightarrow X_j$. Assume that the given TSP cycles in $G[X_i]$ and $G[X_j]$ are $T_{X_i} = x_0...x_{n-1}$ and $T_{X_j} = y_0...y_{n-1}$, respectively. Then, $X_i = \{x_0, ..., x_{n-1}\}$ and $X_j = \{y_0, ..., y_{n-1}\}$. For each $0 \leq k \leq p-1$, define the regular-game set

$$m_k = \{x_{i'} \rightarrow y_{(i'+k) \bmod n}\}_{i'=0}^{n-1}. \tag{1}$$

Moreover, let $\overline{m}_k$ denote the set of the regular-games in m_k with reversed venues, i.e., $\overline{m}_k = \{x_{i'} \leftarrow y_{(i'+k) \bmod n}\}_{i'=0}^{n-1}$. Then, $\bigcup_{k=0}^{n-1}(m_k \cup \overline{m}_k) = \{x_{i'} \leftrightarrow y_{j'} \mid x_{i'} \in X_i, y_{j'} \in X_j\}$.

We define a *block* of regular-games played over 6 consecutive days between all regular-teams in $X_i \cup X_j$ as

$$M_k = m_{3k-3}m_{3k-2}m_{3k-1}\overline{m_{3k-3}m_{3k-2}m_{3k-1}}, \tag{2}$$

for each $1 \leq k \leq \lfloor \frac{n}{3} \rfloor$.

Next, we describe how to schedule all regular-games in the set $\bigcup_{k=0}^{n-1}(m_k \cup \overline{m}_k)$, depending on the value of $n \bmod 3$.

Case 1: $n \bmod 3 = 0$. The regular-games are scheduled over $2n$ days as follows:

$$M_1...M_{\frac{n}{3}}.$$

Case 2: $n \bmod 3 = 1$**.** The regular-games are scheduled over $2n$ days as follows:

$$M_1 ... M_{\frac{n-4}{3}} m_{n-4} m_{n-3} \overline{m_{n-4} m_{n-3}} m_{n-2} m_{n-1} \overline{m_{n-2} m_{n-1}}.$$

Case 3: $n \bmod 3 = 2$**.** The regular-games are scheduled over $2n$ days as follows:

$$M_1 ... M_{\frac{n-2}{3}} m_{n-2} m_{n-1} \overline{m_{n-2} m_{n-1}}.$$

For instance, when $n = 4$, the regular-games are arranged in the order of $m_0 m_1 \overline{m_0 m_1} m_2 m_3 \overline{m_2 m_3}$, spanning 8 days.

We can observe that when a super-team plays two consecutive away normal super-games, in the extension, every regular-team within the super-team does not play more than three consecutive home (or away) regular-games. Moreover, in the extension, whenever a regular-team plays consecutive away regular-games, the locations of the opponents appear consecutively in the TSP cycle T_{X_i} or T_{X_j}.

Left super-games: $X_i \xrightarrow{\mathbf{L}} X_j$**.** Let $T_{X_i} = x_0 ... x_{n-1}$ and $T_{X_j} = y_0 ... y_{n-1}$, and we define m_i and M_i in the same way. Then, we schedule all regular-games in $\bigcup_{k=0}^{n-1}(m_k \cup \overline{m}_k)$.

Case 1: $n \bmod 3 = 0$**.** The regular-games are scheduled over $2n$ days as follows:

$$M_1 ... M_{\frac{n-3}{3}} m_{n-3} m_{n-2} \overline{m_{n-1} m_{n-3} m_{n-2}} m_{n-1}$$

Case 2: $n \bmod 3 = 1$**.** The regular-games are scheduled over $2n$ days as follows:

$$M_1 ... M_{\frac{n-4}{3}} m_{n-4} m_{n-3} \overline{m_{n-4} m_{n-3}} m_{n-2} \overline{m_{n-1} m_{n-2}} m_{n-1}.$$

Case 3: $n \bmod 3 = 2$**.** The regular-games are scheduled over $2n$ days as follows:

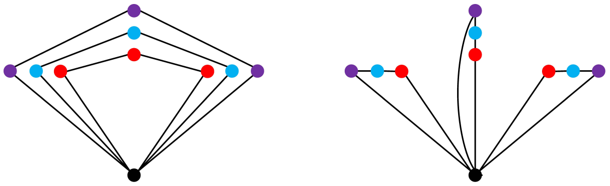
$$M_1 ... M_{\frac{n-2}{3}} m_{n-2} \overline{m_{n-1} m_{n-2}} m_{n-1}.$$

The extension of left super-games is designed this way for the following reason. In the SRR tournament, each super-team plays $p-1$ super-games over $p-1$ time slots, with its home or away status unchanged unless it participates in a left super-game. That is, if a super-team plays an away (resp., home) normal super-game in one time slot, it will continue to play an away (resp., home) super-game in the next. However, if it plays an away (resp., home) left super-game, it will switch to a home (resp., away) game in the next.

The Case where p is Odd

In this case, n is even by definition. If we continue to treat each league as a super-team, then an SRR tournament cannot be constructed since p is odd. To address this issue, we transform each TSP cycle T_{X_i} in $G[X_i]$ into two disjoint cycles of size $n/2$, denoted by $T^a_{X_i}$ and $T^b_{X_i}$, respectively. The details of this transformation are also deferred to the algorithm description in the next section. Then, we let $X^a_i = V(T^a_{X_i})$ and $X^b_i = V(T^b_{X_i})$, and refer to them as *half-leagues*.

Let $\widetilde{\mathcal{X}} = \{X^a_i, X^b_i \mid i \in [p]\}$. In this subsection, we refer to each half-league in $\widetilde{\mathcal{X}}$ as a super-team. Thus, we now have $2p$ super-teams, which is even. Then, similar to the case where p is even, we first design an SRR tournament for the super-teams. Then, we obtain a solution to the p-partite TTP by extending the super-games into regular-games.



(a) Our solution (b) An optimal solution

Figure 4: An illustration of the intra-league property, where vertices (regular-teams) in the same league share the same color: whenever the black vertex plays three consecutive away regular-games, its opponents are from the same league (resp., different leagues) in our (resp., an optimal) solution.

Note that for each $i \in [p]$, the half-leagues X^a_i and X^b_i originate from the same league X_i. Hence, in the SRR tournament construction, we must not arrange a super-game between the super-teams X^a_i and X^b_i for each $i \in [p]$. This property can be ensured via the following labeling method.

Label $Y_i = X^a_i$ and $Y_{2p+1-i} = X^b_i$ for each $i \in [p]$, and denote the resulting set of super-teams as $\{Y_1, Y_2, ..., Y_{2p}\}$.

We then reuse the previous SRR tournament on this set. Note that in this tournament, all super-games between Y_i and Y_{2p+1-i} (for each $i \in [p]$) are scheduled in the last time slot (see Fig. 2). Thus, by simply omitting these super-games, we obtain a new valid SRR tournament that avoids super-games between the two half-leagues of the same original league.

Finally, we extend each super-game in the modified SRR tournament into regular-games using the same method, as in the even-p case, thereby obtaining a feasible solution.

Remark 1. *When $p > 2$, our schedule satisfies the* intra-league *property: when a regular-team plays two or three consecutive away regular-games, its opponents are from the same league, say X_i, and the locations of these opponents appear consecutively along a cycle in $G[X_i]$. This property may be a limitation, as in an optimal solution to the p-partite TTP, a regular-team may achieve a much smaller travel cost by playing consecutive away regular-games against opponents from different leagues (see Fig. 4). Currently, it remains unknown how to construct a high-quality schedule that does not rely on the intra-league property for $p > 2$.*

Performance of the Schedule

We first prove the feasibility of our schedule

Lemma 3 (*)**.** *For the p-partite TTP with any $p \geq 2$, our schedule construction computes a feasible solution.*

Remark 2. *If we replace the left super-game $X_p \xrightarrow{L} X_{p-1}$ in the first time slot with a normal super-game $X_p \leftarrow X_{p-1}$, and the left super-game $X_p \xrightarrow{L} X_1$ in the last time slot with a normal super-game $X_p \rightarrow X_1$, our schedule construction still produces a feasible solution. We may adapt this modification in our experiments.*

To analyze the total travel cost of all regular-teams in our schedule, we make the following assumption.

Assumption 1. *For each super-game, all involved regular-teams are at home before the first regular-game of the super-game begins and return home after the last one ends.*

By the triangle inequality, we can observe that Assumption 1 does not decrease the travel cost of any regular-team. Moreover, it enables us to analyze the total travel cost of all regular-teams by evaluating the travel cost incurred in each super-game separately and then summing the results.

Lemma 4 (*). *Assume that there exists a super-game between super-teams Z_i and Z_j, and let the TSP cycles in $G[Z_i]$ and $G[Z_j]$ be T_{Z_i} and T_{Z_j}, respectively. Then, there exists an $O(n^3)$-time algorithm to ensure that the total travel cost of all regular-teams for playing the extended regular-games is at most $\frac{4n+32}{3n}c(Z_i, Z_j) + \frac{2n}{3}(c(T_{Z_i}) + c(T_{Z_j}))$.*

Lemma 4 will be used to analyze the cost of our schedule.

The Algorithm and Analysis

We first show how to compute the TSP cycle T_{X_i} in $G[X_i]$ for each $i \in [p]$ and transform each T_{X_i} into two disjoint cycles $T^a_{X_i}$ and $T^b_{X_i}$ of size $n/2$ when p is odd. Then, through a novel analysis, we prove an approximation ratio of $\frac{8}{3}+O(\frac{1}{n})$ for the 3-partite TTP. For any $V' \subseteq V$, in what follows, we let MST(V') (resp., TSP(V')) denote the cost of a minimum cost spanning tree (resp., TSP cycle) in $G[V']$.

Computing the Cycles

For the problem of computing a minimum cost TSP cycle in a metric graph, there exists a well-known 1.5-approximation algorithm (Christofides 2022). However, to find the TSP cycle T_{X_i} in $G[X_i]$, instead of directly applying this algorithm to $G[X_i]$, we first compute a TSP cycle T'_{X_i} in $G[X_i \cup \{x^*\}]$ using the algorithm in (Christofides 2022), where $x^* \in \overline{X}_i$ is chosen so that the cost of T'_{X_i} is minimized. This choice is essential for analyzing the approximation ratio of our algorithm. Then, we obtain the desired TSP cycle T_{X_i} in $G[X_i]$ by removing x^* from T'_{X_i}.

When p is odd, we further transform T_{X_i} into two disjoint cycles $T^a_{X_i}$ and $T^b_{X_i}$ of size $n/2$ as follows.

Assume $T_{X_i} = x_0x_1...x_{n-1}$. Note that by deleting the edges $x_{i'}x_{i'+1}$ and $x_{i'+n/2}x_{i'+1+n/2}$ and adding the edges $x_{i'}x_{i'+1+n/2}$ and $x_{i'+1}x_{i'+n/2}$ w.r.t. T_{X_i}, where $1 \le i' \le n$ and we let $x_{i''} = x_{i''-n}$ for each $i'' \ge n$, we can obtain two cycles of size $n/2$, denoted by $T^a_{X_i}(i')$ and $T^b_{X_i}(i')$.

We choose $T^a_{X_i} = T^a_{X_i}(i^*)$ and $T^b_{X_i} = T^b_{X_i}(i^*)$, where $i^* \in [n]$ satisfies that $c(T^a_{X_i}(i^*)) + c(T^b_{X_i}(i^*))$ is minimized.

The details for computing the cycles are put in Alg. 1. We remark that the algorithm in (Christofides 2022) takes $O(n^3)$ time. Thus, Alg. 1 takes $O(p^2n^4)$ time.

By the triangle inequality, it is easy to obtain that $c(T^a_{X_i}) + c(T^b_{X_i}) \le 2c(T_{X_i})$. However, this simple result may worsen the approximation ratio of our algorithm for odd p. Thus, we obtain a better bound on $c(T^a_{X_i}) + c(T^b_{X_i})$.

Lemma 5 (*). *For any $i \in [p]$, it holds that $c(T^a_{X_i}) + c(T^b_{X_i}) \le c(T_{X_i}) + \frac{4}{(p-1)n^2}c(X_i, \overline{X}_i)$.*

Now, we can use Lemmas 4 and 5 to analyze our schedule.

Algorithm 1: Compute the Cycles

```
Require: The graph $G = (V, E)$.
Ensure: A set of cycles.
for each $n$-team league $X_i \in \mathcal{X}$ do
    Compute a TSP cycle $T'_{X_i}$ in $G[X_i \cup \{x^*\}]$ using the algorithm in (Christofides 2022), where $x^* \in \overline{X}_i$ is chosen so that the cost of $T'_{X_i}$ is minimized.
    Obtain the TSP cycle $T_{X_i}$ in $G[X_i]$ by removing the vertex $x^*$ from $T'_{X_i}$.
    if $p$ is odd then
        Obtain the cycles $T^a_{X_i}(i')$ and $T^b_{X_i}(i')$ for each $i' \in [n]$. Let $T^a_{X_i} = T^a_{X_i}(i^*)$ and $T^b_{X_i} = T^b_{X_i}(i^*)$, where $i^* = \arg\min_{i' \in [n]} c(T^a_{X_i}(i')) + c(T^b_{X_i}(i'))$.
    end if
end for
```

Lemma 6 (*). *For the p-partite TTP, there is an $O(p^2n^4)$-time algorithm that computes a schedule with cost at most $\sum_{X_i \in \mathcal{X}}(\frac{2n+24}{3n}c(X_i, \overline{X}_i) + \frac{2(p-1)n}{3}c(T_{X_i}))$.*

Analyzing the Approximation Ratio

First, using a similar argument as in Lemma 2, we have

Lemma 7 (*). *For any $i \in [p]$ and any $x \in X_i$, it holds that $\frac{2}{3}c(x, \overline{X}_i) \le OPT(x) \le 2c(x, \overline{X}_i)$ and $TSP(\overline{X}_i \cup \{x\}) \le OPT(x)$. Moreover, we have $\sum_{X_i \in \mathcal{X}} \frac{2}{3}c(X_i, \overline{X}_i) \le OPT$.*

Next, we focus on breaking the 3-approximation barrier for the 3-partite TTP. Lemma 7 also provides an upper bound on $\sum_{X_i \in \mathcal{X}} c(X_i, \overline{X}_i)$. By Lemma 6, we still need an upper bound on $\sum_{X_i \in \mathcal{X}} c(T_{X_i})$. We first prove one key lemma.

Lemma 8 (*). *For any $i \in [p]$, it holds that $c(T_{X_i}) \le \min_{x \in \overline{X}_i}(MST(X_i \cup \{x\}) + \frac{1}{2}OPT(x))$.*

We remark that if we only use the fact that the algorithm in (Christofides 2022) achieves an approximation ratio of 1.5, we may only obtain $c(T_{X_i}) \le \min_{x \in \overline{X}_i} \frac{3}{2}\text{OPT}(x)$, which yields a *weaker bound* than Lemma 8. Then, we are ready to obtain an upper bound on the item $\sum_{X_i \in \mathcal{X}} c(T_{X_i})$.

Lemma 9 (*). *It holds that $\sum_{X_i \in \mathcal{X}} c(T_{X_i}) \le \frac{p+2}{2(p-1)n}OPT$.*

By Lemmas 2, 6, 7, and 9, we obtain the following result.

Theorem 2. *For the p-partite TTP with any fixed $p \ge 2$, there exists a $\min\{\frac{p+5}{3} + \frac{12}{n}, 3\}$-approximation algorithm with a running time of $O(p^2n^4)$.*

For the 3-partite TTP, our algorithm achieves an approximation ratio of $\frac{8}{3} + O(\frac{1}{n})$. Note that our analysis focuses on the case $p \ge 3$. Although the approximation ratio of our algorithm achieves $\frac{7}{3} + O(\frac{1}{n})$ for $p = 2$, it may be further improved to $\frac{5}{3}+O(\frac{1}{n})$, similar to the TSP-based $(\frac{5}{3}+O(\frac{1}{n}))$-approximation algorithm for TTP (Yamaguchi et al. 2011).

Experimental Results

To generate instances for the p-partite TTP, we adapt the well-known benchmark instances of TTP (Trick 2025; Bulck et al. 2020) by *sequentially* partitioning the teams into p

leagues of equal size. The sizes of the TTP instances range from 4 to 40 teams. We focus on the instance sets GAL, NFL, NL, SUP, and BRA: the GAL set is derived from 3D locations of exoplanets, while the other four are based on real-world sports scheduling scenarios, such as the National Football League, the National League of Major League Baseball, the Super 14 Rugby cup, and the Brazilian soccer championship. There are other TTP instance sets, such as CON, where the distance between every pair of teams is uniform, but these are too special to be considered in our experiments. By our schedule construction, when p is even, we use TTP instances where the total number of teams N satisfies $N \bmod p = 0$ and $\frac{N}{p} \geq 2$; when p is odd, $N \bmod 2p = 0$ and $\frac{N}{2p} \geq 2$. As a result, we test 42 instances for $p = 2$, 10 instances for $p = 3$, and 20 instances for $p = 4$.

To test the performance of our algorithm, we first compare our results with the *Independent Lower Bound* (ILB) (Easton, Nemhauser, and Trick 2002; Urrutia, Ribeiro, and Melo 2007), a well-known approach for deriving lower bounds for TTP. The ILB is computed by determining the minimum possible travel cost for each team independently of the feasibility constraints of the other teams, and then summing these costs. Hence, for the p-partite TTP, computing the minimum possible travel cost for a single team $x \in X_i$ reduces to solving an instance of 3-TC (Asano et al. 1997), where each $x' \in \overline{X}_i$ is treated as a client and x is treated as the root. For the 2-partite TTP, since the $(\frac{3}{2}+\varepsilon)$-approximation algorithm in (Zhao and Xiao 2024) is difficult to implement, we also compare our algorithm with the $(2+\varepsilon)$-approximation algorithm in (Hoshino and Kawarabayashi 2013), which requires the league size n to satisfy $n \bmod 3 = 0$.

To implement our algorithm efficiently, instead of using Alg.1 to compute the cycles, we adopt the well-known 2-opt heuristic for the TSP because it is simple and often produces significantly better solutions than Christofides' algorithm (Christofides 2022) on real-world instances (Bentley 1992). Since a normal super-game may incur a smaller travel cost than a left super-game, we also apply the two modifications described in Remark 2 to replace the two left super-games with normal super-games. Moreover, when labeling (half-)leagues as super-teams, we consider all possible labelings and return the best solution found. Specifically, when p is even, we label the leagues $\mathcal{X} = \{X_i \mid i \in [p]\}$ as p super-teams, resulting in $p!$ possible labelings. When p is odd, we label the half-leagues $\widetilde{\mathcal{X}} = \{X_i^a, X_i^b \mid i \in [p]\}$ as $2p$ super-teams in $\{Y_1, ..., Y_{2p}\}$, where Y_i and Y_{2p+1-i} originate from the same league, resulting in $2^p \cdot p!$ possible labelings.

The algorithms are implemented in C++ and executed on a desktop computer with an AMD Ryzen 5 PRO 4650G with Radeon Graphics (3.7 GHz, 32 GB RAM). The algorithms run fast: all instances are solved together within 0.8s overall.

Fig. 5 summarizes our results, showing the approximation gaps of our algorithm for the p-partite TTP with $p = 2, 3, 4$, alongside the previous algorithm in (Hoshino and Kawarabayashi 2013), which applies only to $p = 2$ with $n \bmod 3 = 0$. The vertical axis lists instances, and the horizontal axis shows the gap to the ILB: $\frac{\text{The Result}-\text{The ILB}}{\text{The ILB}}$.

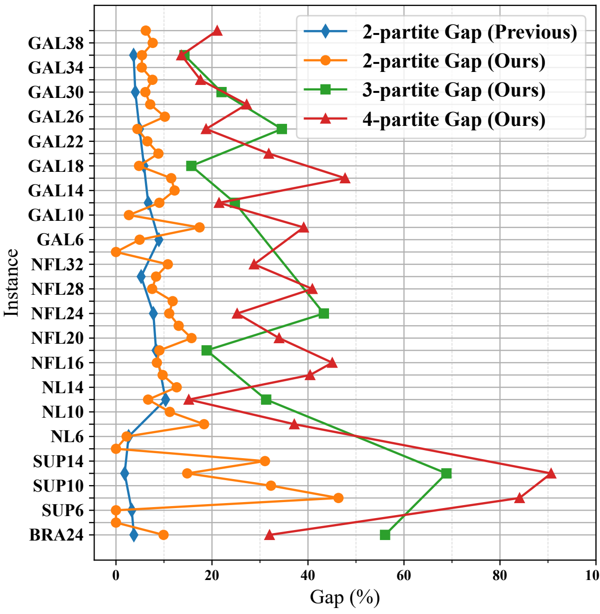


Figure 5: An illustration of the gaps of our algorithms for the p-partite TTP with $p = 2, 3, 4$, as well as those of the previous algorithm for $p = 2$ when $n \bmod 3 = 0$.

We observe that for the case $p = 2$, the gap of our results falls within $5\% - 10\%$ for most instances; for $p = 3$, the gap does not exceed 70% for any tested instance; and for $p = 4$, the gap lies within $35\% - 45\%$ for most instances. Note that our algorithm has a slightly larger gap on the SUP set, where the total number of teams N is small. We further remark that the average gaps for our algorithm are 10.23%, 32.97%, and 35.60% for $p = 2, 3, 4$, respectively. For the case $p = 2$ with $n \bmod 3 = 0$, our gap reaches 6.92%, compared to 5.52% achieved by the algorithm in (Hoshino and Kawarabayashi 2013). Detailed results for each instance and each p are put in the full version. Since ILB $\leq$ OPT, the actual quality of our solutions is much closer to optimality than what the approximation factor guarantees.

Conclusion

In this paper, we introduce the p-partite TTP, a natural generalization of BTTP with distinct structural properties. We prove its NP-hardness for any fixed $p \geq 3$, and propose a TSP-based algorithm that constructs feasible solutions and breaks the 3-approximation barrier for $p = 3$. The algorithm also performs well in experiments.

Several directions remain for future research.

From a theoretical standpoint, it remains open whether the 3-approximation barrier can be broken for any fixed $p > 3$. One promising direction is to design algorithms that avoid relying on intra-league structure. It is also worthwhile to explore the hardness of approximation, such as establishing APX-hardness for the p-partite TTP. On the practical side, it would be interesting to incorporate more heuristic techniques to further improve solution quality.

Acknowledgments

JZ and MX are supported by the National Natural Science Foundation of China under the grants 62502078 and 62372095, and by the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20251102. KK is supported by JSPS Kakenhi 22H05001, and by JST ASPIRE JPMJAP2302.

References

Asano, T.; Katoh, N.; Tamaki, H.; and Tokuyama, T. 1997. Covering Points in the Plane by k-Tours: Towards a Polynomial Time Approximation Scheme for General k. In Leighton, F. T.; and Shor, P. W., eds., *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, 275–283. ACM.

Benoist, T.; Laburthe, F.; and Rottembourg, B. 2001. Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems. In *Proceedings CPAIOR*, volume 1, 15–26.

Bentley, J. L. 1992. Fast Algorithms for Geometric Traveling Salesman Problems. *INFORMS J. Comput.*, 4(4): 387–411.

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. *Eur. J. Oper. Res.*, 280(2): 568–580.

Chatterjee, D.; and Roy, B. K. 2021. An Improved Scheduling Algorithm for Traveling Tournament Problem with Maximum Trip Length Two. In Müller-Hannemann, M.; and Perea, F., eds., *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2021, September 9-10, 2021, Lisbon, Portugal (Virtual Conference)*, volume 96 of *OASIcs*, 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Christofides, N. 2022. Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. *Oper. Res. Forum*, 3(1).

Dantzig, G. B.; and Ramser, J. H. 1959. The truck dispatching problem. *Manag. Sci.*, 6(1): 80–91.

De Werra, D. 1981. Scheduling in sports. *Studies on Graphs and Discrete Programming*, 11: 381–395.

Durán, G. 2021. Sports scheduling and other topics in sports analytics: a survey with special reference to Latin America. *Top*, 29(1): 125–155.

Easton, K.; Nemhauser, G. L.; and Trick, M. A. 2001. The Traveling Tournament Problem Description and Benchmarks. In Walsh, T., ed., *Principles and Practice of Constraint Programming - CP 2001, 7th International Conference, CP 2001, Paphos, Cyprus, November 26 - December 1, 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 Burke, E. K.; and Causmaecker, P. D., eds., *Practice and Theory of Automated Timetabling IV, 4th International Conference, PATAT 2002, Gent, Belgium, August 21-23, 2002, Selected Revised Papers*, volume 2740 of *Lecture Notes in Computer Science*, 100–112. Springer.

Frohner, N.; Neumann, B.; Pace, G.; and Raidl, G. R. 2023. Approaching the Traveling Tournament Problem with Randomized Beam Search. *Evol. Comput.*, 31(3): 233–257.

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

Goerigk, M.; Hoshino, R.; Kawarabayashi, K.; and Westphal, S. 2014. Solving the Traveling Tournament Problem by Packing Three-Vertex Paths. In Brodley, C. E.; and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, 2271–2277. AAAI Press.

Goerigk, M.; and Westphal, S. 2016. A combined local search and integer programming approach to the traveling tournament problem. *Ann. Oper. Res.*, 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, July 22-26, 2007, Vancouver, British Columbia, Canada*, 267–271. AAAI Press.

Hoshino, R.; and Kawarabayashi, K. 2011a. The Inter-League Extension of the Traveling Tournament Problem and its Application to Sports Scheduling. In Burgard, W.; and Roth, D., eds., *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 977–984. AAAI Press.

Hoshino, R.; and Kawarabayashi, K. 2011b. Scheduling Bipartite Tournaments to Minimize Total Travel Distance. *J. Artif. Intell. Res.*, 42: 91–124.

Hoshino, R.; and Kawarabayashi, K. 2012a. Generating Approximate Solutions to the TTP using a Linear Distance Relaxation. *J. Artif. Intell. Res.*, 45: 257–286.

Hoshino, R.; and Kawarabayashi, K. 2012b. The Linear Distance Traveling Tournament Problem. In Hoffmann, J.; and Selman, B., eds., *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, 1770–1778. AAAI Press.

Hoshino, R.; and Kawarabayashi, K. 2013. An Approximation Algorithm for the Bipartite Traveling Tournament Problem. *Math. Oper. Res.*, 38(4): 720–728.

Imahori, S. 2021. A 1+O(1/N) approximation algorithm for TTP(2). *CoRR*, abs/2108.08444.

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

Kendall, G.; Knust, S.; Ribeiro, C. C.; and Urrutia, S. 2010. Scheduling in sports: An annotated bibliography. *Comput. Oper. Res.*, 37(1): 1–19.

Li, M.; Bulck, D. V.; and Goossens, D. R. 2025. Beyond leagues: A single incomplete round robin tournament for multi-league sports timetabling. *Eur. J. Oper. Res.*, 323(1): 208–223.

Li, M.; Davari, M.; and Goossens, D. R. 2023. Multi-league sports scheduling with different leagues sizes. *Eur. J. Oper. Res.*, 307(1): 313–327.

Li, M.; and Goossens, D. R. 2025. Grouping and scheduling multiple sports leagues: an integrated approach. *J. Oper. Res. Soc.*, 76(4): 739–757.

Lim, A.; Rodrigues, B.; and Zhang, X. 2006. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *Eur. J. Oper. Res.*, 174(3): 1459–1478.

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

Thielen, C.; and Westphal, S. 2011. Complexity of the traveling tournament problem. *Theor. Comput. Sci.*, 412(4-5): 345–351.

Thielen, C.; and Westphal, S. 2012. Approximation algorithms for TTP(2). *Math. Methods Oper. Res.*, 76(1): 1–20.

Trick, M. 2025. Challenge traveling tournament instances. https://mat.tepper.cmu.edu/TOURN/. Accessed: 2025-07-25.

Urrutia, S.; Ribeiro, C. C.; and Melo, R. A. 2007. A New Lower Bound to the Traveling Tournament Problem. In *2007 IEEE Symposium on Computational Intelligence in Scheduling, CISched 2007, Honolulu, Hawaii, USA, April 2-4, 2007*, 15–18. IEEE, IEEE.

Westphal, S.; and Noparlik, K. 2014. A 5.875-approximation for the Traveling Tournament Problem. *Ann. Oper. Res.*, 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 Faliszewski, P.; Muscholl, A.; and Niedermeier, R., eds., *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, 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. 2023. The Linear Distance Traveling Tournament Problem Allows an EPTAS. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 12155–12162. AAAI Press.

Zhao, J.; and Xiao, M. 2024. A Better Approximation for Bipartite Traveling Tournament in Inter-League Sports Scheduling. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 6814–6822. ijcai.org.

Zhao, J.; and Xiao, M. 2025a. The APX-hardness of the Traveling Tournament Problem. *Oper. Res. Lett.*, 62: 107311.

Zhao, J.; and Xiao, M. 2025b. A matching-based approximation algorithm for the traveling tournament problem. *Theor. Comput. Sci.*, 1055: 115485.

Zhao, J.; and Xiao, M. 2025c. Practical Algorithms with Guaranteed Approximation Ratio for Traveling Tournament Problem with Maximum Tour Length 2. *Math. Oper. Res.*, 50(2): 910–934.

Zhao, J.; Xiao, M.; and Xu, C. 2022. Improved Approximation Algorithms for the Traveling Tournament Problem. In Szeider, S.; Ganian, R.; and Silva, A., eds., *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs*, 83:1–83:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.