Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)

A Better Approximation for Bipartite Traveling Tournament in Inter-League Sports Scheduling

Jingyang Zhao and **Mingyu Xiao***
University of Electronic Science and Technology of China, Chengdu, China
{jingyangzhao1020, myxiao}@gmail.com

Abstract

The bipartite traveling tournament problem (BTTP) was initially introduced by Hoshino and Kawarabayashi (AAAI 2011) to address inter-league sports scheduling, which aims to design a feasible bipartite tournament between two n-team leagues under some constraints such that the total traveling distance of all participating teams is minimized. Since its introduction, several methods have been developed to design feasible schedules for NBA, NPB and so on. In terms of solution quality with a theoretical guarantee, previously only a $(2+\varepsilon)$ approximation is known for the case that $n \equiv 0 \pmod 3$. Whether there are similar results for the cases that $n \equiv 1 \pmod 3$ and $n \equiv 2 \pmod 3$ was asked in the literature. In this paper, we answer this question positively by proposing a $(3/2+\varepsilon)$-approximation algorithm for any n and any constant $\varepsilon > 0$, which also improves the previous ratio for the case that $n \equiv 0 \pmod 3$.

1 Introduction

The traveling tournament problem (TTP), introduced by Easton *et al.* [2001], is a well-known benchmark problem in the field of sports scheduling [Kendall *et al.*, 2010]. This problem aims to find a double round-robin tournament, minimizing the total traveling distance of all participating teams. In a double round-robin tournament involving n teams (where n is even), each team plays two games against each of the other $n-1$ teams that includes one home game at its own home venue and one away game at its opponent's home venue, and on each day each team can only play one game. Moreover, all games should be scheduled on $2(n-1)$ consecutive days, subject to several constraints on the maximum number of consecutive home/away games for each team, ensuring a balanced tournament arrangement. In the problem, we may also consider that the distance is a semi-metric, i.e., it satisfies the symmetry and triangle inequality properties. An overview of TTP and its variants, along with their various applications on sports scheduling, can be found in [Bulck *et al.*, 2020; Durán, 2021].

*Contact Author

The bipartite traveling tournament problem (BTTP), introduced by Hoshino and Kawarabayashi [2011b], is an inter-league extension of TTP. Given two n-team leagues, it asks for a distance-optimal double round-robin bipartite tournament between the leagues, where every team in one league plays one home game and one away game against each team in another league. It also requires that each team plays only one game on each day, and all games need to be scheduled on $2n$ consecutive days. For TTP/BTTP, the double round-robin (bipartite) tournament is subject to the following three basic constraints or assumptions:

- *No-repeat*: No pair of teams can play against each other in two consecutive games.
- *direct-traveling*: Each team travels directly from its game venue on the i-th day to its game venue on the $(i+1)$-th day, where we assume that all teams are initially at home before the first game starts and will return home after the last game ends.
- *Bounded-by-3*: Each team can play at most 3-consecutive home games or away games.

In the last constraint, we require that the maximum number of consecutive home/away games for each team is at most three. This is the most extensively studied case [Lim *et al.*, 2006; Anagnostopoulos *et al.*, 2006; Goerigk *et al.*, 2014]. The case that at most two consecutive home/away games are allowed is also studied in some references [Thielen and Westphal, 2012; Chatterjee and Roy, 2021; Zhao and Xiao, 2021b]. A small number of consecutive home/away games requires teams to return home frequently, which may make the tournament balanced as a cost of a possible longer traveling distance.

As indicated in a previous study [Hoshino and Kawarabayashi, 2011c], BTTP has natural applications in sports scheduling, such as for events like the Davis Cup, the biennial Ryder Cup, the National Basketball Association (NBA), and the Nippon Professional Baseball (NPB). Exploring BTTP also holds promise for advancing the theoretical development of enhanced algorithms for TTP. For instance, Zhao and Xiao [2023] utilized an approximation algorithm for BTTP between two groups by grouping teams and achieved an efficient polynomial-time approximation scheme for a special case of TTP where all teams are positioned along a line. BTTP possesses a simpler structure, and scheduling for TTP can be derived from scheduling for

BTTP using a divide-and-conquer method [Chatterjee and Roy, 2021]. Therefore, effective algorithms for BTTP have the potential to yield effective algorithms for TTP as well.

1.1 Related Work

Both TTP and BTTP are difficult optimization problems, and their NP-hardness has been established in [Thielen and Westphal, 2011; Hoshino and Kawarabayashi, 2011b]. In the online benchmark [Trick, 2024; Bulck *et al.*, 2020], many instances of TTP with more than ten teams have not been completely solved even by using high-performance machines. TTP and BTTP have been extensively studied both in theory [Hoshino and Kawarabayashi, 2012; Hoshino and Kawarabayashi, 2013; Westphal and Noparlik, 2014; Xiao and Kou, 2016; Imahori, 2021; Zhao and Xiao, 2021a] and practice [Easton *et al.*, 2002; Di Gaspero and Schaerf, 2007; Hentenryck and Vergados, 2007; Hoshino and Kawarabayashi, 2011a; Hoshino and Kawarabayashi, 2011b; Hoshino and Kawarabayashi, 2011c; Goerigk and Westphal, 2016; Frohner *et al.*, 2023].

TTP and BTTP are also rich problems in approximation algorithms. An algorithm is called an α-approximation algorithm if it can generate a feasible schedule in polynomial time such that the total traveling distance of all teams is within α times the optimal. For TTP, Miyashiro *et al.* [2012] first proposed a $(2+\varepsilon)$-approximation algorithm where ε is an arbitrary fixed constant, and the ratio was later improved to $(1.667+\varepsilon)$ by Yamaguchi *et al.* [2011], and $(1.598+\varepsilon)$ by Zhao *et al.* [2022]. For BTTP, the only known result is a $(2+\varepsilon)$-approximation algorithm for the case that $n \equiv 0 \pmod 3$, proposed by Hoshino and Kawarabayashi [2013], and whether there exist similar algorithms for the cases that $n \equiv 1 \pmod 3$ and $n \equiv 2 \pmod 3$ was asked in this paper.

1.2 Our Results

In this paper, we design a new algorithm for BTTP and prove that our algorithm can achieve an approximation ratio of $(3/2+\varepsilon)$ for any n and any constant $\varepsilon > 0$. This not only positively answers the open question for the cases that $n \equiv 1 \pmod 3$ and $n \equiv 2 \pmod 3$ but also improves the previous best ratio of $(2+\varepsilon)$ for the case that $n \equiv 0 \pmod 3$ [Hoshino and Kawarabayashi, 2013].

To achieve our improvement, we introduce a novel lower bound for BTTP related to the minimum weight cycle packing (i.e., a set of cycles and each cycle has a length of at least 3). Then, we propose a 3-path packing construction, which reduces BTTP to the task of finding a good 3-path packing. At last, we use the minimum weight cycle packing to obtain a 3-path packing, and show that by applying the 3-path packing to our construction we can get a schedule with an approximation ratio of at most $(3/2+\varepsilon)$ in polynomial time.

In addition to the theoretical results, we also consider the practical applications of our construction. We create a new instance from the real situation of NBA for BTTP with 32 teams ($n = 16$) and test our algorithm on this instance. Note that previous construction only works for the case that $n \equiv 0 \pmod 3$. In the new instance, $n \equiv 1 \pmod 3$. We extend our 3-path packing construction to a more efficient construction based on 3-cycles. By applying these two constructions to the new instance, experimental results show that our algorithms can be implemented within 2 seconds, and the quality of the obtained solutions is much better than the expected 1.5-approximation ratio: the gaps between our results and the optimal are at most 24.66% and 9.42% for the 3-path and 3-cycle constructions.

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.

2 Notations

An instance of BTTP can be presented by a complete graph $G = (V = X \cup Y, E, w)$ with $2n$ vertices representing $2n$ teams, where $X = \{x_0, \ldots, x_{n-1}\}$ and $Y = \{y_0, \ldots, y_{n-1}\}$ are two n-team leagues, and w is a non-negative semi-metric weight function on the edges in E that satisfies the symmetry and triangle inequality properties, i.e., $w(x, y) + w(y, z) \geq w(x, z) = w(z, x)$ for any $x, y, z \in V$. The weight $w(x, y)$ of edge $xy \in E$ represents the distance between the homes of teams x and y. We also extend the function to a set of edges, i.e., we let $w(E') := \sum_{e \in E'} w(e)$ for any $E' \subseteq E$, and to a subgraph G' of G, i.e., we let $w(G')$ be the total weight of all edges in G'. Given any $V' \subseteq V$, the complete graph induced by V' is denoted by $G[V']$.

Given two vertices/teams $x \in X$ and $y \in Y$, and two sets $X' \subseteq X$ and $Y' \subseteq Y$, we define the following notations. We use $x \to y$ to denote a game between x and y at the home of y, and $x \leftrightarrow y$ to denote two games between x and y including one game at the home of x and one game at the home of y. We use $E_{Y'}(x)$ (resp., $E_{X'}(y)$) to denote the set of edges in G between the vertex x (resp., y) and one vertex in Y' (resp., X'), i.e., $E_{Y'}(x) = \{xy \mid y \in Y'\}$. We also let $\delta_{Y'}(x) = w(E_{Y'}(x))$ and $\delta_{Y'}(X') = \sum_{x \in X'} \delta_{Y'}(x)$ (resp., $\delta_{X'}(y) = w(E_{X'}(y))$ and $\delta_{X'}(Y') = \sum_{y \in Y'} \delta_{X'}(y)$). Note that $\delta_{Y'}(X') = \delta_{X'}(Y')$.

Two subgraphs or sets of edges are *vertex-disjoint* if they do not share a common vertex. An l-cycle $x_1x_2 \ldots x_lx_1$ is a simple cycle on l different vertices $\{x_1, \ldots, x_l\}$. It consists of l edges $\{x_1x_2, \ldots, x_lx_1\}$, and its *length* is said to be l. A cycle packing in a graph is a set of vertex-disjoint cycles, where the length of each cycle is at least three and the cycles cover all vertices of the graph. The minimum weight cycle packing in a graph with n vertices can be found in $O(n^3)$ time [Hartvigsen, 1984]. Similarly, an l-path $x_1x_2 \ldots x_l$ is a simple path on l different vertices $\{x_1, \ldots, x_l\}$. It consists of $l-1$ edges $\{x_1x_2, \ldots, x_{l-1}x_l\}$, and the vertices x_1 and x_l are called its *terminals*. An *l-path packing* in a graph is a set of vertex-disjoint l-paths, where the paths cover all vertices of the graph.

A *walk* is a sequence of vertices where each consecutive pair of vertices is connected by an edge, and its weight is defined as the total weight of the edges traversed in the sequence. A walk is *closed* if the first and the last vertices are the same. In a solution of BTTP, every team $v \in X \cup Y$ has an *itinerary*, which is a closed walk starting and ending at v. This itinerary could be decomposed into several minimal closed walks, each starting and ending at v, referred to as *trips*. Each trip is an l-cycle containing v with $2 \leq l \leq 4$, and

	0	1	2	3	4	5
x_0	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$
x_1	$\boldsymbol{y_2}$	y_0	y_1	y_2	$\boldsymbol{y_0}$	$\boldsymbol{y_1}$
x_2	$\boldsymbol{y_1}$	$\boldsymbol{y_2}$	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
y_1	x_2	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0	x_1
y_2	x_1	x_2	$\boldsymbol{x_0}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	x_0

Table 1: A solution for BTTP with two leagues $X = \{x_0, x_1, x_2\}$ and $Y = \{y_0, y_1, y_2\}$, where home games are marked in bold

all trips/cycles share only one common vertex v. For example, Table 1 shows a solution for leagues $X = \{x_0, x_1, x_2\}$ and $Y = \{y_0, y_1, y_2\}$, where home games are marked in bold. We can get that x_2 has an itinerary $x_2y_0y_1y_2x_2$, and y_2 has an itinerary $y_2x_1x_2y_2x_0y_2$, and y_2 has two trips $y_2x_1x_2y_2$ and $y_2x_0y_2$ (a 3-cycle and a 2-cycle sharing y_2 only).

Fix a constant $\varepsilon > 0$. We want to consider a 3-path packing in G. Since n may not be divisible by 3, we will remove a small number of vertices (as we will see that the number is related to ε) in X and Y to obtain two new sets $X_\varepsilon \subseteq X$ and $Y_\varepsilon \subseteq Y$ such that the number of vertices in X_ε and Y_ε, denoted as n_ε, are the same and can be divisible by 3. Hence, there always exist 3-path packings in $G[X_\varepsilon]$ and $G[Y_\varepsilon]$. We also let $\overline{X_\varepsilon} := X \setminus X_\varepsilon$ and $\overline{Y_\varepsilon} := Y \setminus Y_\varepsilon$. The graph $G_\varepsilon := G[X_\varepsilon \cup Y_\varepsilon]$ is called the *core* graph of G because we will see that the quality of our schedule is only dominated by the total traveling distance of teams in $X_\varepsilon \cup Y_\varepsilon$, i.e., the total traveling distance of teams in $\overline{X_\varepsilon} \cup \overline{Y_\varepsilon}$ is small. We denote the minimum weight cycle packing in $G[X_\varepsilon]$ (resp., $G[Y_\varepsilon]$) by $\mathcal{C}_{X_\varepsilon}$ (resp., $\mathcal{C}_{Y_\varepsilon}$).

3 Lower Bounds

Lower bounds play an important role in approximation algorithms. We need to compare our solution with the optimal solution. However, it is hard to compute the optimal solution. Then, we turn to find some lower bounds of the optimal value and compare our solution with the lower bounds. We will use two lower bounds. The first one is from the literature and the second one is newly proved. We use OPT to denote the weight (i.e., the total traveling distance of all teams) of an optimal solution for BTTP.

Lemma 1. [Hoshino and Kawarabayashi, 2013]. *For BTTP, it holds that* $2\delta_X(Y) = \delta_Y(X) + \delta_X(Y) \leq \frac{3}{2} \cdot \mathit{OPT}$.

Next, we will propose a new lower bound that is related to the minimum weight cycle packings $\mathcal{C}_{X_\varepsilon}$ and $\mathcal{C}_{Y_\varepsilon}$.

Lemma 2 (*). *For BTTP, it holds that* $\delta_{Y_\varepsilon}(X_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{Y_\varepsilon}) + \delta_{X_\varepsilon}(Y_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{X_\varepsilon}) \leq \frac{3}{2} \cdot \mathit{OPT}$.

4 The Construction of the Schedule

Assuming we are given a 3-path packing $\mathcal{P}_{X_\varepsilon}$ in $G[X_\varepsilon]$ and a 3-path packing $\mathcal{P}_{Y_\varepsilon}$ in $G[Y_\varepsilon]$, to construct a schedule that minimizes the total traveling distance of teams in $X_\varepsilon \cup Y_\varepsilon$, the idea is to ensure every team in X_ε (res., Y_ε) plays 3-consecutive away games along every 3-path in $\mathcal{P}_{Y_\varepsilon}$ (resp., $\mathcal{P}_{X_\varepsilon}$) from one terminal to another.

Consider an *ideal* schedule where every team in Y_ε (resp., X_ε) plays 3-consecutive away games along every 3-path in $\mathcal{P}_{X_\varepsilon}$ (resp., $\mathcal{P}_{Y_\varepsilon}$). Then, every team in Y_ε (resp., X_ε) plays $\frac{n_\varepsilon}{3}$ away-trips with teams in X_ε (resp., Y_ε). The total traveling distance of teams in Y_ε for these away-trips is

$$\sum_{y \in Y_\varepsilon} \sum_{\substack{P = xx'x'' \\ \in \mathcal{P}_{X_\varepsilon}}} (w(y,x) + w(y,x'') + w(P)) = \sum_{xx'x'' \in \mathcal{P}_{X_\varepsilon}} (\delta_{Y_\varepsilon}(x) + \delta_{Y_\varepsilon}(x'')) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}).$$

Denote $\sum_{xx'x'' \in \mathcal{P}_{X_\varepsilon}} (\delta_{Y_\varepsilon}(x) + \delta_{Y_\varepsilon}(x''))$ by $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon})$ for the sake of presentation and define $\delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon})$ in the similar manner. The total traveling distance of teams in $X_\varepsilon \cup Y_\varepsilon$ for these away-trips is $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon})$.

Our construction will generate a schedule such that the total traveling distance of teams in $X \cup Y$ is close to $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon})$, i.e, the performance of our schedule is close to the ideal schedule. To achieve this, we make sure that almost every team in Y_ε (resp., X_ε) plays 3-consecutive away games along every 3-path in $\mathcal{P}_{X_\varepsilon}$ (resp., $\mathcal{P}_{Y_\varepsilon}$). Next, we give our construction in details.

Note that $\varepsilon > 0$ is a fixed constant. Let $d := 6\lceil 1/\varepsilon \rceil$ and $m := 2\lfloor \frac{n}{6d} \rfloor - 1$ (it is useful to ensure that d is even and m is odd). We will select $n_\varepsilon := 3md$ vertices from X (resp., Y) to form X_ε (resp., Y_ε). The details of how to select these vertices is deferred to the analysis part of our schedule. Recall that $\mathcal{P}_{X_\varepsilon}$ and $\mathcal{P}_{Y_\varepsilon}$ are two given 3-path packings in $G[X_\varepsilon]$ and $G[Y_\varepsilon]$. So, both $\mathcal{P}_{X_\varepsilon}$ and $\mathcal{P}_{Y_\varepsilon}$ contain md 3-paths. Assume that $\mathcal{P}_{X_\varepsilon} = \{P_1, \ldots, P_{md}\}$ and $\mathcal{P}_{Y_\varepsilon} = \{Q_1, \ldots, Q_{md}\}$, where $P_i = x_{3i-3}x_{3i-2}x_{3i-1}$ and $Q_i = y_{3i-3}y_{3i-2}y_{3i-1}$. Each 3-path corresponds to three teams. For every d 3-paths from P_1 (resp., Q_1) to P_{md} (resp., Q_{md}), we pack the corresponding $3d$ teams as a *super-team*, denoted by S_i (resp., T_i). So, $S_i = \{x_{3i'-3}, x_{3i'-2}, x_{3i'-1}\}_{i'=id-d}^{id-1}$. Let $l := n - n_\varepsilon$. There are l teams in $\overline{X_\varepsilon}$ (resp., $\overline{Y_\varepsilon}$), and we label them using $\{x_{n_\varepsilon}, \ldots, x_{n-1}\}$ (resp., $\{y_{n_\varepsilon}, \ldots, y_{n-1}\}$). We can get l *team-pairs* $\{L_1, \ldots, L_l\}$ where $L_i = \{x_{n_\varepsilon+i-1}, y_{n_\varepsilon+i-1}\}$.

We may assume that $n \geq 108d^3$, as otherwise, we have $n < 108d^3 = O_\varepsilon(1)$[1], and in this case, we can solve the problem optimally in constant time. Since $n \geq 108d^3$, we can get $m \geq 18d^2$. Moreover, since $l = n - n_\varepsilon = n - 3md = n - 6d\lfloor \frac{n}{6d} \rfloor + 3d$, we can get $3d \leq l \leq 9d \leq m$.

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 BTTP.

Our schedule contains $m + 1$ time slots, where we have m super-games in each of the first m time slots, including l *left super-games* and $m - l$ *normal super-games*. Each super-team will attend one super-game in each of the first m time slots. The last time slot is special, and we will explain it later. Each of the first m time slots spans $6d$ days and the last time slot spans $2l$ days. Hence, our schedule will span $6md + 2l = 2n_\varepsilon + 2l = 2n$ days.

[1] $O_\varepsilon(1)$ means a constant related to ε.

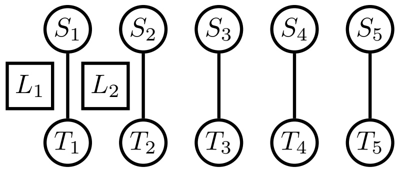

Figure 1: The super-game schedule in the first time slot, where $m = 5$ and $l = 2$

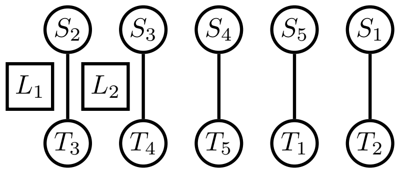

Figure 2: The super-game schedule in the second slot, where $m = 5$ and $l = 2$

4.1 The First m Time Slots

In the first time slot, the super-games are arranged as shown in Figure 1, where $m = 5$ and $l = 2$. Each super-team is denoted by a cycle node and each team-pair is denoted by a square node. There are $2m$ super-teams, l team-pairs, and m super-games. Each super-game is denoted by an edge between two super-teams. The most left l super-games, each involving two super-teams and one team-pair, are called the *left super-games*. The left super-game involving team-pair L_i is denoted as the *i-th left super-game*. The other $m - l$ super-games are called *normal super-games*. It is worth noting that there are only a constant number of left super-games since $l \leq 9d = O_{\varepsilon}(1)$. So, in our construction almost all super-games are normal super-games.

In the second time slot, the super-games are arranged as shown in Figure 2. Compared to the first time slot, the positions take one leftward shift for super-teams $S_1, \ldots, S_m$, take two leftward shifts for super-teams $T_1, \ldots, T_m$, and keep unchanged for team-pairs $L_1, \ldots, L_l$.

The schedules for the first m time slots are derived analogously. We have two observations.

Claim 3. *During the first m time slots, there is only one super-game between super-teams S_i and T_j for every $1 \leq i, j \leq m$.*

Proof. By the construction, there is a super-game between super-teams S_i and T_j if and only if the schedule is at the time slot of $(j - i + 1 + m) \bmod m$. □

Claim 4. *During the first m time slots, each super-team participates in only one i-th left super-game for every $1 \leq i \leq r$.*

Proof. From the 1-st to the m-th time slot, the super-team in $\{S_1, \ldots, S_m\}$ playing an i-th left super-game is $S_i, S_{i+1}, \ldots, S_m, S_1, \ldots, S_{i-1}$, respectively; the super-team in $\{T_1, \ldots, T_m\}$ playing an i-th left super-game is $T_i, T_{i+2}, \ldots, T_{i-1}, T_{i+1}, \ldots, T_{i-2}$, respectively, since m is odd. So, each super-team plays only one i-th left super-game. □

Next, we show how to extend normal super-games and left super-games into normal games.

	0	1	2	3	4	5	6	7	8	9	10	11
s_0	t_0	t_1	t_2	$\boldsymbol{t_0}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	t_4	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$
s_1	$\boldsymbol{t_5}$	t_0	t_1	t_2	$\boldsymbol{t_0}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	t_4	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$
s_2	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_0	t_1	t_2	$\boldsymbol{t_0}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	t_4	t_5	$\boldsymbol{t_3}$
s_3	t_3	t_4	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_0	t_1	t_2	$\boldsymbol{t_0}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$
s_4	$\boldsymbol{t_2}$	t_3	t_4	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_0	t_1	t_2	$\boldsymbol{t_0}$	$\boldsymbol{t_1}$
s_5	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	t_4	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_0	t_1	t_2	$\boldsymbol{t_0}$
t_0	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0	s_1	s_2	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3	s_4	s_5
t_1	s_5	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0	s_1	s_2	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3	s_4
t_2	s_4	s_5	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0	s_1	s_2	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3
t_3	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3	s_4	s_5	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0	s_1	s_2
t_4	s_2	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3	s_4	s_5	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0	s_1
t_5	s_1	s_2	$\boldsymbol{s_3}$	$\boldsymbol{s_4}$	$\boldsymbol{s_5}$	s_3	s_4	s_5	$\boldsymbol{s_0}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	s_0

Table 2: Extending the normal super-game between $\{s_0, s_1, s_2, s_3, s_4, s_5\}$ and $\{t_0, t_1, t_2, t_3, t_4, t_5\}$ into normal games on $6d = 12$ days, where $d = 2$ and home games are marked in bold

We consider a super-game between super-teams S_i and T_j. For the sake of presentation, we define $s_k := x_{(i-1)d+k}$ and $t_k := y_{(j-1)d+k}$ for any $0 \leq k < 3d$. By definitions of S_i and T_j, we have that $S_i = \{s_0, s_1, s_2, \ldots, s_{3d-3}, s_{3d-2}, s_{3d-1}\}$ and $T_j = \{t_0, t_1, t_2, \ldots, t_{3d-3}, t_{3d-2}, t_{3d-1}\}$.

First consider that the super-game is a normal super-game.

Normal super-games: The normal super-game will be extended into normal games on $6d$ days in the following way:

- Team $s_{3i+i'}$ plays an away game with $t_{3j+j'}$ on $(6(i + j) + i' + j') \bmod 6d$ day,
- Team $s_{3i+i'}$ plays a home game with $t_{3j+j'}$ on $(6(i + j) + i' + j' + 3) \bmod 6d$ day,

where $0 \leq i, j \leq d - 1$ and $0 \leq i', j' \leq 2$. An illustration of the normal games after extending one normal super-game for $d = 2$ is shown in Table 2.

The design of normal super-games is essential the same of the construction in [Hoshino and Kawarabayashi, 2013], which only works for the case that $n \equiv 0 \pmod 3$. It guarantees that all games between one team in S_i and one team in T_j are arranged. To get a feasible schedule for the cases that $n \equiv 1 \pmod 3$ and $n \equiv 2 \pmod 3$, we need to design left super-games, which are newly introduced.

Left super-games: In this case, we have a team-pair $L_{i'}$, so we also define $s_{3d} := x_{n_{\varepsilon}+i'-1}$ and $t_{3d} := y_{n_{\varepsilon}+i'-1}$. By definition of $L_{i'}$, we can get $L_{i'} = \{s_{3d}, t_{3d}\}$. For all teams in $S_i \cup T_j \cup R_{i'}$, we define a set of games as

$$m_i := \{s_{i'} \to t_{(i'+i) \bmod (3d+1)}\}_{i'=0}^{3d},$$

where each team in $S_i \cup T_j \cup R_{i'}$ plays one game. We also define $\overline{m_i}$ as the set of games in m_i but with reversed venues, and let $\overline{\overline{m_i}} := m_i$. Since d is even, the extended normal games on $6d$ days can be presented by

$$m_1\overline{m_2}m_3\overline{m_4}\cdots m_{3d-1}\overline{m_{3d}} \cdot \overline{\overline{m_2}m_3\overline{m_4}m_5\cdots\overline{m_{3d}}m_1}.$$

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

Note that if d is odd, the games can be presented by

$$m_1\overline{m_2}m_3\overline{m_4}\cdots m_{3d} \cdot \overline{m_1\overline{m_2}m_3\overline{m_4}\cdots m_{3d}}.$$

For each left super-game, we have two more teams due to a team-pair, and then two days' games in m_0 and $\overline{m_0}$ are still unarranged in extending the super-game on $6d$ days.

	0	1	2	3	4	5	6	7	8	9	10	11
s_0	t_1	$\boldsymbol{t_2}$	t_3	$\boldsymbol{t_4}$	t_5	$\boldsymbol{t_6}$	t_2	$\boldsymbol{t_3}$	t_4	$\boldsymbol{t_5}$	t_6	$\boldsymbol{t_1}$
s_1	t_2	$\boldsymbol{t_3}$	t_4	$\boldsymbol{t_5}$	t_6	$\boldsymbol{t_0}$	t_3	$\boldsymbol{t_4}$	t_5	$\boldsymbol{t_6}$	t_0	$\boldsymbol{t_2}$
s_2	t_3	$\boldsymbol{t_4}$	t_5	$\boldsymbol{t_6}$	t_0	$\boldsymbol{t_1}$	t_4	$\boldsymbol{t_5}$	t_6	$\boldsymbol{t_0}$	t_1	$\boldsymbol{t_3}$
s_3	t_4	$\boldsymbol{t_5}$	t_6	$\boldsymbol{t_0}$	t_1	$\boldsymbol{t_2}$	t_5	$\boldsymbol{t_6}$	t_0	$\boldsymbol{t_1}$	t_2	$\boldsymbol{t_4}$
s_4	t_5	$\boldsymbol{t_6}$	t_0	$\boldsymbol{t_1}$	t_2	$\boldsymbol{t_3}$	t_6	$\boldsymbol{t_0}$	t_1	$\boldsymbol{t_2}$	t_3	$\boldsymbol{t_5}$
s_5	t_6	$\boldsymbol{t_0}$	t_1	$\boldsymbol{t_2}$	t_3	$\boldsymbol{t_4}$	t_0	$\boldsymbol{t_1}$	t_2	$\boldsymbol{t_3}$	t_4	$\boldsymbol{t_6}$
s_6	t_0	$\boldsymbol{t_1}$	t_2	$\boldsymbol{t_3}$	t_4	$\boldsymbol{t_5}$	t_1	$\boldsymbol{t_2}$	t_3	$\boldsymbol{t_4}$	t_5	$\boldsymbol{t_0}$
t_0	$\boldsymbol{s_6}$	s_5	$\boldsymbol{s_4}$	s_3	$\boldsymbol{s_2}$	s_1	$\boldsymbol{s_5}$	s_4	$\boldsymbol{s_3}$	s_2	$\boldsymbol{s_1}$	s_6
t_1	$\boldsymbol{s_5}$	s_4	$\boldsymbol{s_3}$	s_2	$\boldsymbol{s_1}$	s_0	$\boldsymbol{s_4}$	s_3	$\boldsymbol{s_2}$	s_1	$\boldsymbol{s_0}$	s_5
t_2	$\boldsymbol{s_4}$	s_3	$\boldsymbol{s_2}$	s_1	$\boldsymbol{s_0}$	s_6	$\boldsymbol{s_3}$	s_2	$\boldsymbol{s_1}$	s_0	$\boldsymbol{s_6}$	s_4
t_3	$\boldsymbol{s_3}$	s_2	$\boldsymbol{s_1}$	s_0	$\boldsymbol{s_6}$	s_5	$\boldsymbol{s_2}$	s_1	$\boldsymbol{s_0}$	s_6	$\boldsymbol{s_5}$	s_3
t_4	$\boldsymbol{s_2}$	s_1	$\boldsymbol{s_0}$	s_6	$\boldsymbol{s_5}$	s_4	$\boldsymbol{s_1}$	s_0	$\boldsymbol{s_6}$	s_5	$\boldsymbol{s_4}$	s_2
t_5	$\boldsymbol{s_1}$	s_0	$\boldsymbol{s_6}$	s_5	$\boldsymbol{s_4}$	s_3	$\boldsymbol{s_0}$	s_6	$\boldsymbol{s_5}$	s_4	$\boldsymbol{s_3}$	s_1
t_6	$\boldsymbol{s_0}$	s_6	$\boldsymbol{s_5}$	s_4	$\boldsymbol{s_3}$	s_2	$\boldsymbol{s_6}$	s_5	$\boldsymbol{s_4}$	s_3	$\boldsymbol{s_2}$	s_0

Table 3: Extending the left super-game between $\{s_0, s_1, s_2, s_3, s_4, s_5\}$ and $\{t_0, t_1, t_2, t_3, t_4, t_5\}$ into normal games on $6d = 12$ days, where $d = 2$ and home games are marked in bold

4.2 The Last Time Slot

In the last time slot, we will arrange all unarranged games due to left super-games. First, we will consider the unarranged games involving teams in team-pairs. Then, we will consider the unarranged games involving teams in super-teams.

The teams in team-pairs. Recall that the l team-pairs contain l teams in $\overline{X_\varepsilon} = \{x_{n_\varepsilon}, \dots, x_{n-1}\}$ and l teams in $\overline{Y_\varepsilon} = \{y_{n_\varepsilon}, \dots, y_{n-1}\}$. The next claim shows the unarranged normal games involving teams in team-pairs.

Claim 5 (*). *The unarranged normal games involving teams in team-pairs are* $\{x_i \leftrightarrow y_j \mid x_i \in \overline{X_\varepsilon}, y_j \in \overline{Y_\varepsilon}\}$.

The games in $\{x_i \leftrightarrow y_j \mid x_i \in \overline{X_\varepsilon}, y_j \in \overline{Y_\varepsilon}\}$ forms a bipartite tournament between two l-team leagues $\overline{X_\varepsilon}$ and $\overline{Y_\varepsilon}$. Next, we design a simple algorithm to arrange them.

Let $s_i := x_{n_\varepsilon+i}$ and $t_i := y_{n_\varepsilon+i}$ for any $0 \leq i < l$. We define a set of games as

$$m_i := \{s_{i'} \to t_{(i'+i) \bmod l}\}_{i'=0}^{l-1}.$$

The games on $2l$ days for $\overline{X_\varepsilon} \cup \overline{Y_\varepsilon}$ can be presented by

$$m_0\overline{m_1}m_2\overline{m_3}\cdots\overline{m_0\overline{m_1}m_2\overline{m_3}\cdots}.$$

Since $l \geq 3d$, we can avoid the infeasible case $m_0\overline{m_0}$ $(l = 1)$.

The teams in super-teams. By Claim 4, for any $1 \leq i \leq l$, each super-team plays only one i-th left super-game. Hence, there are m i-th left super-games, denoted as $\{S_i - T_{j_i}\}_{i=1}^{m}$. Recall that in the i-th left super-game two days' games in $m_0 \cup \overline{m_0}$ were not arranged. We denote the union of unarranged games (involving teams in super-teams) in m_i for all m i-th left super-games as M_i. Then, the unarranged games (involving teams in super-teams) due to the left super-games are $\bigcup_{i=1}^{l} M_i \cup \overline{M_i}$. The games on $2l$ days for $X_\varepsilon \cup Y_\varepsilon$ can be presented by

$$M_1\overline{M_2}M_3\overline{M_4}\cdots\overline{M_1\overline{M_2}M_3\overline{M_4}\cdots}.$$

Theorem 6 (*). *For BTTP with $n \geq 108d^3$, the above construction generates a feasible solution.*

Although the above theorem requires n being a large number, it can guarantee our expected approximation ratio since an instance with a constant n can be optimally solved in constant time. However, for the sake of application, we also show that our construction can be slightly modified to work for instances with almost all n.

Theorem 7 (*). *For BTTP, our construction can be modified slightly to generate a feasible solution for any n except for $n \in \{1, 2, 5, 8, 14\}$.*

4.3 The Quality of the Schedule

The construction provides a feasible solution if each team in $X \cup Y$ has a label. There are $n! \times n!$ ways to label teams in X using $\{x_0, \dots, x_{n-1}\}$ and teams in Y using $\{y_0, \dots, y_{n-1}\}$. To get a good solution, we label them randomly.

Step 1. Assign each vertex $x \in X$ (resp., $y \in Y$) with a cost of $\delta_Y(x)$ (resp., $\delta_X(y)$), and select the top n_ε vertices with the highest costs to form X_ε (resp., Y_ε).
Step 2. Generate a random permutation of md 3-paths in $\mathcal{P}_{X_\varepsilon}$ (resp., $\mathcal{P}_{Y_\varepsilon}$), and label them as $P_1, \dots, P_{md}$ (resp., $Q_1, \dots, Q_{md}$), respectively, where $P_i = x_{3i-3}x_{3i-2}x_{3i-1}$ and $Q_i = y_{3i-3}y_{3i-2}y_{3i-1}$.
Step 3. Take an arbitrary permutation of l teams in $\overline{X_\varepsilon}$ (resp., $\overline{Y_\varepsilon}$), and label them as $x_{n_\varepsilon}, \dots, x_{n-1}$ (resp., $y_{n_\varepsilon}, \dots, y_{n-1}$), respectively.

Note that Step 1 can be done in $O(n^2)$ time, and Step 2 and Step 3 can be done in $O(n)$ time. In Step 2, we assume the two 3-path packings are given in advance. Moreover, when labeling a 3-path using $P_i = x_{3i-3}x_{3i-2}x_{3i-1}$, there may be two choices, and either of them is considered valid.

Theorem 8. *For BTTP with $n \geq 108d^3$, using the above randomized labeling method, our construction can generate a solution with an expected weight of at most $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) + \varepsilon \cdot OPT$ in $O(n^2)$ time.*

Proof. Clearly, our construction takes $O(n^2)$ time.

Then, we make two assumptions for the sake of analysis, which do not decrease the weight of our schedule by the triangle inequality. Firstly, we assume that every participant team returns home both before and after each day's game in left super-games and the last time slot. Secondly, we assume that all participant teams return home before the game on the first day and after the game on the last day in normal super-games.

By Step 1 of our labeling algorithm, since we select the top n_ε vertices from X with the highest costs to form X_ε, we get

$$\delta_Y(X_\varepsilon) = \sum_{x \in X_\varepsilon} \delta_Y(x) \geq \frac{n_\varepsilon}{n} \sum_{x \in X} \delta_Y(x) = \frac{n_\varepsilon}{n} \delta_Y(X).$$

Alternatively, since $l = n - n_\varepsilon \leq 9d$, we have

$$\delta_{\overline{X_\varepsilon}}(Y) = \delta_Y(\overline{X_\varepsilon}) \leq \left(1 - \frac{n_\varepsilon}{n}\right) \delta_Y(X) \leq \frac{9d}{n} \delta_Y(X). \quad (1)$$

Similarly, we have

$$\delta_{\overline{Y_\varepsilon}}(X) = \delta_X(\overline{Y_\varepsilon}) \leq \left(1 - \frac{n_\varepsilon}{n}\right) \delta_X(Y) \leq \frac{9d}{n} \delta_X(Y). \quad (2)$$

Consider a team $x \in \overline{X_\varepsilon}$. By assumptions, in our schedule the weight of its itinerary is $\sum_{y \in Y} 2w(x, y) = 2\delta_Y(x)$. The total weight of itineraries of teams in $\overline{X_\varepsilon}$ is $\sum_{x\in\overline{X_\varepsilon}} 2\delta_Y(x) = 2\delta_Y(\overline{X_\varepsilon}) \leq \frac{18d}{n}\delta_Y(X) \leq \frac{1}{d}\delta_Y(X)$ by (1) and $n \geq 108d^3$.

Fix a team $x \in S_i \subseteq X_\varepsilon$. We have three parts of its trips.

1. The weight of its trips for playing teams in $\overline{Y_\varepsilon}$ is $\sum_{y\in\overline{Y_\varepsilon}} 2w(x,y) = 2\delta_{\overline{Y_\varepsilon}}(x)$ by assumptions since these games are in left super-games.
2. In a left super-game between S_i and T_j, the weight of its trips for playing teams in T_j is $\sum_{y\in T_j} 2w(x,y) = 2\delta_{T_j}(x)$. We can get $\mathbb{E}[\delta_{T_j}(x)] = \frac{1}{m}\delta_{Y_\varepsilon}(x)$ by Step 2 of our labeling algorithm. Since S_i plays l left super-games in total, the expected weight of its trips for playing teams in a super-team that plays a left super-game with S_i is $\frac{2l}{m}\delta_{Y_\varepsilon}(x) \leq \frac{1}{d}\delta_{Y_\varepsilon}(x)$ due to $l \leq 9d$ and $m \geq 18d^2$.
3. In a normal super-game between S_i and T_j, it plays d or $d+1$ trips along at least $d-1$ 3-paths in $\mathcal{Q}_j := \{Q_{(j-1)d+1}, \ldots, Q_{jd}\}$ from one terminal to another (see Table 2). If it plays d trips along d 3-paths in $\mathcal{Q}_j$, the weight of these trips is $\delta_x(\mathcal{Q}_j) + w(\mathcal{Q}_j)$; if it plays $d+1$ trips along $d-1$ 3-paths in $\mathcal{Q}_j$, it does not follow a 3-path $y'_0y'_1y'_2$ in $\mathcal{Q}_j$, the weight of the two trips for playing teams in $\{y'_0, y'_1, y'_2\}$ is bounded by $2w(x,y'_0) + 2w(x,y'_1) + 2w(x,y'_2)$ by the triangle inequality, and then the weight of the $d+1$ trips is bounded by $\delta_x(\mathcal{Q}_j) + w(\mathcal{Q}_j) + 2w(x,y'_0) + 2w(x,y'_1) + 2w(x,y'_2)$. We have $\mathbb{E}[\delta_x(\mathcal{Q}_j)] = \frac{1}{m}\delta_x(\mathcal{P}_{Y_\varepsilon})$, $\mathbb{E}[w(\mathcal{Q}_j)] = \frac{1}{m}w(\mathcal{P}_{Y_\varepsilon})$, and $\mathbb{E}[w(x,y'_0) + w(x,y'_1) + w(x,y'_2)] = \frac{1}{md}\delta_{Y_\varepsilon}(x)$ by Step 2 of our labeling algorithm. Since S_i plays $m-l \leq m$ normal super-games, the expected weight of trips for playing teams in a super-team that plays a normal super-game with S_i is at most $\delta_x(\mathcal{P}_{Y_\varepsilon}) + w(\mathcal{P}_{Y_\varepsilon}) + \frac{1}{d}\delta_{Y_\varepsilon}(x)$.

Hence, the total expected weight of itineraries of teams in X_ε is at most

$$\begin{aligned} &2\delta_{\overline{Y_\varepsilon}}(X_\varepsilon) + \frac{2}{d}\delta_{Y_\varepsilon}(X_\varepsilon) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) \\ &\leq \frac{3}{d}\delta_X(Y) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) \end{aligned}$$

since we have $2\delta_{\overline{Y_\varepsilon}}(X_\varepsilon) \leq 2\delta_{\overline{Y_\varepsilon}}(X) \leq \frac{1}{d}\delta_X(Y)$ by (2) and $n \geq 108d^3$, and $\delta_{Y_\varepsilon}(X_\varepsilon) \leq \delta_Y(X) = \delta_X(Y)$.

Therefore, the total expected weight of itineraries of teams in X is bounded by $\frac{4}{d}\delta_X(Y) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon})$. Similarly, for teams in Y, the expected weight is bounded by $\frac{4}{d}\delta_Y(X) + \delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon})$.

Since $d = 6\lceil 1/\varepsilon\rceil \geq 6/\varepsilon$ by our setting and $\delta_Y(X) = \delta_X(Y) \leq (3/4)\cdot$ OPT by Lemma 1, our schedule has an expected weight of at most $\varepsilon\cdot$OPT$+\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon})$. □

5 The 3-Path Packing

In this section, we will obtain novel 3-path packings $\mathcal{P}_{X_\varepsilon}$ in $G[X_\varepsilon]$ and $\mathcal{P}_{Y_\varepsilon}$ in $G[Y_\varepsilon]$ such that $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) \leq \delta_{Y_\varepsilon}(X_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{X_\varepsilon}) + \delta_{X_\varepsilon}(Y_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{Y_\varepsilon})$. By Lemma 2 and Theorem 8, these two 3-path packings directly imply a $(3/2+\varepsilon)$-approximation for BTTP.

Next, we mainly show how to obtain $\mathcal{P}_{X_\varepsilon}$ in $G[X_\varepsilon]$ such that $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) \leq \delta_{Y_\varepsilon}(X_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{X_\varepsilon})$ since we can get $\mathcal{P}_{Y_\varepsilon}$ in a similar way. We define some notations.

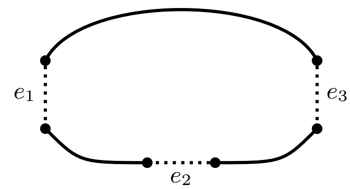

Figure 3: An illustration of patching paths into a cycle: there are three paths, denoted by solid lines, and they are patched into a cycle using three edges e_1, e_2, e_3, denoted by dashed lines

Since $\mathcal{C}_{X_\varepsilon}$ is a cycle packing in $G[X_\varepsilon]$, the length of each cycle in $\mathcal{C}_{X_\varepsilon}$ is at least 3 and at most n_ε. Let $\mathcal{B}_q$ be the set of all q-cycles in $\mathcal{C}_{X_\varepsilon}$. So, we have $\mathcal{C}_{X_\varepsilon} = \bigcup_{q=3}^{n_\varepsilon} \mathcal{B}_q$.

We will use $\mathcal{C}_{X_\varepsilon}$ to obtain a cycle packing $\mathcal{C}$ in $G[X_\varepsilon]$ such that the length of each cycle in $\mathcal{C}$ is divisible by 3, and then use $\mathcal{C}$ to obtain the 3-path packing $\mathcal{P}_{X_\varepsilon}$. The algorithm of $\mathcal{C}$ is shown as follows.

Step 1. Initialize $\mathcal{C} = \emptyset$.
Step 2. For each 3-cycle in $\mathcal{B}_3$, we directly choose it into $\mathcal{C}$.
Step 3. For each q-cycle $C_q \in \mathcal{C}_{X_\varepsilon} \setminus \mathcal{B}_3$, obtain a q-path P_q by deleting the edge $xx' \in C_q$ such that $\delta_{Y_\varepsilon}(x) + \delta_{Y_\varepsilon}(x') - n_\varepsilon w(x,x')$ minimized. Then, patch all paths into a single cycle C arbitrarily, and select the cycle into the packing.

An illustration of the patching operation in Step 3 can be seen in Figure 3. Moreover, it is easy to see that the length of each cycle in $\mathcal{C}$ is divisible by 3, and the algorithm takes $O(n^3)$ time, dominated by computing the minimum cycle packing $\mathcal{C}_{X_\varepsilon}$ [Hartvigsen, 1984].

Lemma 9 (*). $n_\varepsilon w(\mathcal{C}) \leq \frac{1}{2}\delta_{Y_\varepsilon}(X_\varepsilon) + \frac{3}{4}n_\varepsilon w(\mathcal{C}_{X_\varepsilon})$.

Theorem 10 (*). *There is a $O(n^3)$-time algorithm to get a 3-path packing $\mathcal{P}_{X_\varepsilon}$ in $G[X_\varepsilon]$ and a 3-path packing $\mathcal{P}_{Y_\varepsilon}$ in $G[Y_\varepsilon]$ with $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) \leq \delta_{Y_\varepsilon}(X_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{X_\varepsilon})$ and $\delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) \leq \delta_{X_\varepsilon}(Y_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{Y_\varepsilon})$.*

Theorem 11. *For BTTP with any n and any constant $\varepsilon > 0$, there is a randomized $(3/2+\varepsilon)$-approximation algorithm with a running time of $O(n^3)$.*

Proof. By Theorem 10, we can get 3-path packings $\mathcal{P}_{X_\varepsilon}$ in $G[X_\varepsilon]$ and $\mathcal{P}_{Y_\varepsilon}$ in $G[Y_\varepsilon]$ in $O(n^3)$ time with $\delta_{Y_\varepsilon}(\mathcal{P}_{X_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{X_\varepsilon}) + \delta_{X_\varepsilon}(\mathcal{P}_{Y_\varepsilon}) + n_\varepsilon w(\mathcal{P}_{Y_\varepsilon}) \leq \delta_{Y_\varepsilon}(X_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{X_\varepsilon}) + \delta_{X_\varepsilon}(Y_\varepsilon) + \frac{1}{2}n_\varepsilon w(\mathcal{C}_{Y_\varepsilon})$.

If $n \geq 108d^3$, by Lemma 2 and Theorem 8, we can obtain a solution in $O(n^2)$ time with an expected weight of at most $(3/2+\varepsilon)\cdot$ OPT by calling $\mathcal{P}_{X_\varepsilon}$ and $\mathcal{P}_{Y_\varepsilon}$ on our randomized construction algorithm. Otherwise, as mentioned, we have $n < 108d^3 = O_\varepsilon(1)$, and in this case, we can solve the problem optimally in constant time. Therefore, there is a randomized $(3/2+\varepsilon)$-approximation algorithm for BTTP, and the total running time is $O(n^3)$. □

Our algorithm is randomized because it uses a simple randomized labeling technique. It can be derandomized in polynomial time using the well-known method of conditional expectations in [Williamson and Shmoys, 2011], while preserving the approximation ratio.

6 Application

To test the performance of our algorithm, we introduce a new BTTP instance, and apply our algorithm to this instance. Additionally, we extend our construction to a *3-cycle construction* by slightly modifying the design of normal super-games, which turns out to be more practical.

The New Instance. The construction of our instance is motivated by the real situation of NBA. Since 2004, the number of teams in NBA has always been 30, where there are 15 teams in the Western Conference and 15 teams in the Eastern Conference. Thus, Hoshino and Kawarabayashi [2011b] constructed an NBA instance with $n = 15$ for BTTP. In recent years, there have been rumors saying that NBA is poised to expand to 32 teams in 2024, with the potential inclusion of two new teams from Las Vegas and Seattle. Therefore, we create an instance, where we assume that two new teams from Las Vegas and Seattle join in the Western Conference, and also the Minnesota Timberwolves, originally from the Western Conference, are moved to join the Eastern Conference. This is an instance for BTTP with $n = 16$. We determined their distance matrix using the following method: first, we surveyed the home venues of the existing 30 teams and made educated guesses about the home venues of the two new teams; then, we obtained the longitude and latitude coordinates of these home venues using Google Maps; last, we calculated the distances (in miles) between any pair of teams based on the coordinates using the Haversine formula.

The Parameters. We set $d = 1$ and $m = 5$. Then, we have $l = 1$, and the games in the last time slot may violate the no-repeat constraint. As in the proof of Theorem 7, if we denote the games on $2n$ days in our construction as $g_1 g_2 \dots g_{2n}$, we can rearrange them as $g_{2n} g_1 g_2 \dots g_{2n-1}$ to fix this problem.

The 3-cycle Construction. Our construction can be easily extended to pack 3-cycles. We can modify our normal super-games as follows. Assume that there is a normal super-game between $S = \{s_0, s_1, s_2\}$ and $T = \{t_0, t_1, t_2\}$. In the extension, we define $m_i := \{s_{i'} \to t_{(i'+i) \bmod 3}\}_{i'=0}^{2}$, and arrange games on 6 days in the order of $m_0 m_1 m_2 \overline{m_0 m_1 m_2}$.

In the extension of the new normal super-game, each team in S (resp., T) plays 3-consecutive away games along two edges in the 3-cycle $t_0 t_1 t_2 t_0$ (resp., $s_0 s_1 s_2 s_0$). Moreover, on each day teams in the same league either all play home games or all play away games. Due to this, before extending one new normal super-game between super-teams S and T, we may relabel teams in S and teams T. There are $3! \times 3! = 36$ ways, and we choose the best one so that the total traveling distance of teams in $S \cup T$ is minimized on the extended 6 days. We will recover their labels after extending the new normal super-game, and the feasibility still holds. Note that this property does not hold for the previous normal super-game.

Optimizing Left Super-games. The design of left super-games in our construction can be optimized. Recall that we arrange the games in left super-game as $m_1 \overline{m_2} m_3 \overline{m_1} m_2 \overline{m_3}$. To reduce the frequency of returning homes, we rearrange them as $m_1 m_2 \overline{m_3 m_1} m_3 \overline{m_2}$ for the 3-path construction, and $m_1 m_2 m_3 \overline{m_1 m_2 m_3}$ for the 3-cycle construction. The feasibility also holds.

The Implementations. Instead of finding a 3-path packing using the algorithm in Theorem 10, we directly label teams randomly, and try to improve our schedule by exchanging the labels of two teams in the same league. There are $\binom{n}{2}$ pairs of teams in X and $\binom{n}{2}$ pairs of teams in Y. We consider these $n(n-1)$ pairs in a random order. From the first pair to the last, we test whether the weight of our schedule can be reduced after we swap the labels of teams in the pair. If no, we do not swap them and go to the next pair. If yes, we swap them and go to the next pair. After considering all the $n(n-1)$ pairs, if there is an improvement, we repeat the whole procedure. Otherwise, the procedure ends. Since the size of instance is small, it is sufficient to obtain a good solution very quickly.

Our algorithms are coded in C++, on a standard desktop computer with a 3.20GHz AMD Athlon 200GE CPU and 8 GB RAM.

The details of the new instance and our algorithms can be found in https://github.com/JingyangZhao/BTTP.

We compare our results with the well-known *Independent Lower Bound* [Easton *et al.*, 2002]. The idea is to compute the best-possible traveling distance of a single team v (i.e., the traveling distance of its minimum weight itinerary) independently without considering the feasibility of other teams. The value is denoted as ILB_v, and then the independent lower bound is $ILB = \sum_{v \in X \cup Y} ILB_v$. Note that ILB_v can be obtained by solving the Capacitated Vehicle Routing Problem (CVRP). For example, if $v \in X$, we obtain an instance of CVRP, where there is a vehicle with a capacity of 3 at v, each team in Y has a unit demand, and we need to find a minimum itinerary for the vehicle to fulfill the demand of teams in Y. To solve this instance of CVRP, we use a brute-force enumeration since the size is small. We have $ILB = 655475.788$.

Our results can be seen in Table 4, where the column '*Construction*' indicates the 3-path and the 3-cycle constructions; '*Result*' lists the results of our algorithms; '*Gap*' is defined to be $\frac{Result - ILB}{ILB}$, and '*Time*' is the running time of our algorithms. We can see that both of our algorithms run very fast. Compared to ILB, the quality of the 3-path construction has a gap of 24.66%, which is much smaller than the expected 1.5-approximation ratio. Moreover, the 3-cycle construction is more practical, which can reduce the gap to only 9.42%.

Construction	Result	Gap	Time
3-path	817086.814	24.66%	0.85s
3-cycle	717172.797	9.42%	1.03s

Table 4: Experimental results of our constructions

7 Conclusion

In this paper, for BTTP with any n and any constant $\varepsilon > 0$, we propose a $(3/2 + \varepsilon)$-approximation algorithm, which significantly improves the previous result. Our theoretical result relies on three key ideas: the first is a new lower bound, the second is a new construction, and the third is a 3-path packing algorithm. Our methods also have the potential to design better approximation algorithms for TTP, which is left for future work. For applications, we create a new real instance from NBA for BTTP. Experimental results show that the 3-cycle construction has a very good practical performance.

Acknowledgements

This work is supported by National Natural Science Foundation of China under grants 62372095 and 62172077, and Natural Science Foundation of Sichuan Province of China under grant 2023NSFSC0059.

References

[Anagnostopoulos *et al.*, 2006] Aris Anagnostopoulos, Laurent Michel, Pascal Van Hentenryck, and Yannis Vergados. A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2):177–193, 2006.

[Bulck *et al.*, 2020] David Van Bulck, Dries R. Goossens, Jörn Schönberger, and Mario Guajardo. Robinx: A three-field classification and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, 280(2):568–580, 2020.

[Chatterjee and Roy, 2021] Diptendu Chatterjee and Bimal Kumar Roy. An improved scheduling algorithm for traveling tournament problem with maximum trip length two. In *ATMOS 2021*, volume 96, pages 16:1–16:15, 2021.

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

[Durán, 2021] Guillermo Durán. Sports scheduling and other topics in sports analytics: a survey with special reference to latin america. *Top*, 29(1):125–155, 2021.

[Easton *et al.*, 2001] Kelly Easton, George Nemhauser, and Michael Trick. The traveling tournament problem: description and benchmarks. In *CP 2001*, pages 580–584, 2001.

[Easton *et al.*, 2002] Kelly Easton, George L. Nemhauser, and Michael A. Trick. Solving the travelling tournament problem: A combined integer programming and constraint programming approach. In *PATAT 2002*, volume 2740 of *Lecture Notes in Computer Science*, pages 100–109. Springer, 2002.

[Frohner *et al.*, 2023] Nikolaus Frohner, Bernhard Neumann, Giulio Pace, and Günther R Raidl. Approaching the traveling tournament problem with randomized beam search. *Evolutionary Computation*, 31(3):233–257, 2023.

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

[Goerigk *et al.*, 2014] Marc Goerigk, Richard Hoshino, Ken-ichi Kawarabayashi, and Stephan Westphal. Solving the traveling tournament problem by packing three-vertex paths. In *AAAI 2014*, pages 2271–2277. AAAI Press, 2014.

[Hartvigsen, 1984] David Hartvigsen. *Extensions of matching theory*. PhD thesis, Carnegie-Mellon University, 1984.

[Hentenryck and Vergados, 2007] Pascal Van Hentenryck and Yannis Vergados. Population-based simulated annealing for traveling tournaments. In *AAAI 2007*, pages 267–272. AAAI Press, 2007.

[Hoshino and Kawarabayashi, 2011a] Richard Hoshino and Ken-ichi Kawarabayashi. The distance-optimal inter-league schedule for japanese pro baseball. In *Proceedings of the ICAPS 2011 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS)*, pages 71–78, 2011.

[Hoshino and Kawarabayashi, 2011b] Richard Hoshino and Ken-ichi Kawarabayashi. The inter-league extension of the traveling tournament problem and its application to sports scheduling. In *AAAI 2011*, pages 977–984, 2011.

[Hoshino and Kawarabayashi, 2011c] Richard Hoshino and Ken-ichi Kawarabayashi. Scheduling bipartite tournaments to minimize total travel distance. *Journal of Artificial Intelligence Research*, 42:91–124, 2011.

[Hoshino and Kawarabayashi, 2012] Richard Hoshino and Ken-ichi Kawarabayashi. The linear distance traveling tournament problem. In *AAAI 2012*, pages 1770–1778. AAAI Press, 2012.

[Hoshino and Kawarabayashi, 2013] Richard Hoshino and Ken-ichi Kawarabayashi. An approximation algorithm for the bipartite traveling tournament problem. *Mathematics of Operations Research*, 38(4):720–728, 2013.

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

[Kendall *et al.*, 2010] Graham Kendall, Sigrid Knust, Celso C Ribeiro, and Sebastián Urrutia. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, 37(1):1–19, 2010.

[Lim *et al.*, 2006] Andrew Lim, Brian Rodrigues, and Xingwen Zhang. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3):1459–1478, 2006.

[Miyashiro *et al.*, 2012] Ryuhei Miyashiro, Tomomi Matsui, and Shinji Imahori. An approximation algorithm for the traveling tournament problem. *Annals of Operations Research*, 194(1):317–324, 2012.

[Thielen and Westphal, 2011] Clemens Thielen and Stephan Westphal. Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4):345–351, 2011.

[Thielen and Westphal, 2012] Clemens Thielen and Stephan Westphal. Approximation algorithms for TTP(2). *Mathematical Methods of Operations Research*, 76(1):1–20, 2012.

[Trick, 2024] Michael Trick. Challenge traveling tournament instances. Online, 2024.

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

[Williamson and Shmoys, 2011] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[Xiao and Kou, 2016] Mingyu Xiao and Shaowei Kou. An improved approximation algorithm for the traveling tournament problem with maximum trip length two. In *MFCS 2016*, volume 58 of *LIPIcs*, pages 89:1–89:14, 2016.

[Yamaguchi *et al.*, 2011] Daisuke Yamaguchi, Shinji Imahori, Ryuhei Miyashiro, and Tomomi Matsui. An improved approximation algorithm for the traveling tournament problem. *Algorithmica*, 61(4):1077–1091, 2011.

[Zhao and Xiao, 2021a] Jingyang Zhao and Mingyu Xiao. A further improvement on approximating TTP-2. In *COCOON 2021*, volume 13025 of *Lecture Notes in Computer Science*, pages 137–149, 2021.

[Zhao and Xiao, 2021b] Jingyang Zhao and Mingyu Xiao. The traveling tournament problem with maximum tour length two: A practical algorithm with an improved approximation bound. In *IJCAI 2021*, pages 4206–4212, 2021.

[Zhao and Xiao, 2023] Jingyang Zhao and Mingyu Xiao. The linear distance traveling tournament problem allows an EPTAS. In *AAAI 2023*, pages 12155–12162, 2023.

[Zhao *et al.*, 2022] Jingyang Zhao, Mingyu Xiao, and Chao Xu. Improved approximation algorithms for the traveling tournament problem. In *MFCS 2022*, volume 241 of *LIPIcs*, pages 83:1–83:15, 2022.