Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Solving the Traveling Tournament Problem by Packing Three-Vertex Paths

Marc Goerigk
University of Kaiserslautern
Kaiserslautern, Germany

Richard Hoshino
Quest University Canada
Squamish, British Columbia

Ken-ichi Kawarabayashi
National Institute of Informatics
JST ERATO Kawarabayashi
Large Graph Project, Tokyo, Japan

Stephan Westphal
University of Göttingen
Göttingen, Germany

Abstract

The Traveling Tournament Problem (TTP) is a complex problem in sports scheduling whose solution is a schedule of home and away games meeting specific feasibility requirements, while minimizing the total distance traveled by all the teams. A recently-developed "hybrid" algorithm, combining local search and integer programming, has resulted in best-known solutions for many TTP instances. In this paper, we tackle the TTP from a graph-theoretic perspective, by generating a new "canonical" schedule in which each team's three-game road trips match up with the underlying graph's minimum-weight P_3-packing. By using this new schedule as the initial input for the hybrid algorithm, we develop tournament schedules for five benchmark TTP instances that beat all previously-known solutions.

Introduction

Inspired by the real-life problem of optimizing Major League Baseball schedules to reduce team travel, the n-team Traveling Tournament Problem (TTP) asks for the double round-robin schedule that minimizes the sum total of distances traveled by all n teams. Since the problem was first proposed (Easton, Nemhauser, and Trick 2001), the TTP has attracted a significant amount of research (Kendall et al. 2010), with numerous heuristics developed for solving hard TTP instances.

There is an online set of benchmark n-team TTP data sets (Trick 2013) with the list of best-known upper and lower bounds. Solutions to TTP instances are often found after weeks of computation on high-performance machines using parallel computing, see, e.g. (Hentenryck and Vergados 2007). In many ways, the TTP is a variant of the well-known Traveling Salesman Problem (TSP), asking for a distance-optimal schedule linking venues that are close to one another. The computational complexity of the TSP is NP-hard; recently, it was shown that solving the TTP is strongly NP-hard (Thielen and Westphal 2011).

In a recent paper (Goerigk and Westphal 2012), a "hybrid" approach was developed to generate TTP solutions to the instances in the *Galaxy* benchmark data set (Uthus, Riddle, and Guesgen 2012), beating all previously-known upper bounds. Their approach started with a well-known canonical schedule based on an approximate minimum-weight Hamiltonian cycle, followed by a combination of commercial integer programming solvers with local heuristics such as tabu search.

In this paper, we build upon this work, and propose an integrated approach to solving hard TTP instances, consisting of three phases. In Phase 1, a constructive procedure based on the graph-theoretic concept of three-vertex path packings (known as P_3-packings) is used to produce an initial, feasible schedule. In Phase 2, a simple local search procedure known as "pairwise-swapping" attempts to improve this solution. In Phase 3, we take the solution from the previous phase and apply the aforementioned hybrid heuristic of Goerigk and Westphal to output a final solution.

We apply this three-step approach to the n-team Galaxy instances, a data set where the "teams" represent exoplanets located in three-dimensional space. Our method finds best-known solutions to the Galaxy instances for four cases, $n \in \{22, 28, 34, 40\}$.

We also explain how this algorithm can be combined to generate a new solution for the NFL28 data set, beating the previously best-known upper bound (Trick 2013) that was published in 2007.

The main focus of this paper is our explanation of the feasible schedule in Phase One. Our approach is novel in that our schedule is based on a P_3-packing, rather than a TSP-tour (i.e., Hamilton cycle). We will describe the construction of a double round-robin schedule where each team's three-game road trips match up with some (nearly)-optimal P_3-packing. Our n-team tournament schedule is a feasible TTP solution whenever $n = 6m - 2$ for some integer $m \geq 1$.

After describing the three phases, and sharing the results of our new upper bounds, we conclude the paper with some open problems and directions for further research.

The Traveling Tournament Problem

Let $\{t_1, t_2, \ldots, t_n\}$ be the n teams in a sports league, where n is even. Let D be the $n \times n$ *distance matrix*, where entry $D_{i,j}$ is the distance between the home stadiums of teams t_i and t_j. By definition, $D_{i,j} = D_{j,i}$ for all $1 \leq i, j \leq n$, and all diagonal entries $D_{i,i}$ are zero. We assume the distances form a *metric*, i.e., $D_{i,j} \leq D_{i,k} + D_{k,j}$ for all i, j, k.

For example, Table 1 provides the 6×6 distance matrix for the Galaxy set (Trick 2013), consisting of the six "teams" Earth, Eridanus, Ara, Gemini, Pisces, and Cepheus.

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	0	15	22	47	45	50
t_2	15	0	10	34	36	39
t_3	22	10	0	32	31	40
t_4	47	34	32	0	51	41
t_5	45	36	31	51	0	35
t_6	50	39	40	41	35	0

Table 1: The Galaxy6 Distance Matrix.

The TTP requires a tournament lasting $2(n-1)$ days, where every team has exactly one game scheduled each day with no days off (this explains why n must be even.) The objective is to minimize the total distance traveled by the n teams, subject to the following conditions:

(a) *each-venue*: Each pair of teams plays twice, once in each other's home venue.

(b) *at-most-three*: No team may have a home stand or road trip lasting more than three games.

(c) *no-repeat*: A team cannot play against the same opponent in two consecutive games.

When calculating the total distance, we assume that every team begins the tournament at home and returns home after playing its last away game. Furthermore, whenever a team has a road trip consisting of multiple away games, the team doesn't return to their home city but rather proceeds directly to their next away venue.

To illustrate with a specific example, Table 2 lists a feasible tournament schedule for the Galaxy6 benchmark set. In this schedule, as with all subsequent schedules presented in this paper, home games are marked in bold.

Team	1	2	3	4	5	6	7	8	9	10	# of Trips
t_1	t_4	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	t_5	t_6	t_3	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	$\boldsymbol{t_2}$	7
t_2	t_3	$\boldsymbol{t_1}$	$\boldsymbol{t_4}$	t_5	t_6	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	$\boldsymbol{t_3}$	t_1	8
t_3	$\boldsymbol{t_2}$	$\boldsymbol{t_6}$	t_1	t_6	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_1}$	$\boldsymbol{t_4}$	t_2	t_5	7
t_4	$\boldsymbol{t_1}$	$\boldsymbol{t_5}$	t_2	t_1	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_6}$	t_3	t_5	t_6	7
t_5	t_6	t_4	$\boldsymbol{t_6}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	t_2	t_1	$\boldsymbol{t_4}$	$\boldsymbol{t_3}$	7
t_6	$\boldsymbol{t_5}$	t_3	t_5	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_4	t_2	t_1	$\boldsymbol{t_4}$	7

Table 2: An optimal TTP solution for Galaxy6.

For example, team t_1 makes 7 trips, with a travel distance of $D_{1,4} + D_{4,2} + D_{2,1} + D_{1,5} + D_{5,6} + D_{6,3} + D_{3,1} = 238$. We can use Table 1 to show that the total travel distance of this tournament is 1365; it is known that this schedule, with 43 total trips, is distance-optimal (Trick 2013).

Hamiltonian Cycles and P_3-Packings

We motivate our graph-theoretic approach with two separate 15-team instances, which combine to form the 30 teams of the National Basketball Association (NBA). These two-dimensional data sets were introduced in the context of the Bipartite Traveling Tournament Problem (Hoshino and Kawarabayashi 2011), where a close-to-optimal *inter*-league tournament was generated using graph theory. We will now apply these ideas to solve hard instances of the TTP, which models an *intra*-league tournament.

The 30 teams in the NBA are divided into two separate leagues, with 15 in the Western Conference and 15 in the Eastern Conference. Let G_W be the complete graph on 15 vertices, representing the Western Conference teams, with the weight of edge ij being the distance between the home arenas of teams t_i and t_j. Similarly, define G_E to be the graph of the Eastern Conference teams. A computer search quickly finds the minimum-weight Hamiltonian cycle for each 15-team conference, as illustrated in Figure 1 below.

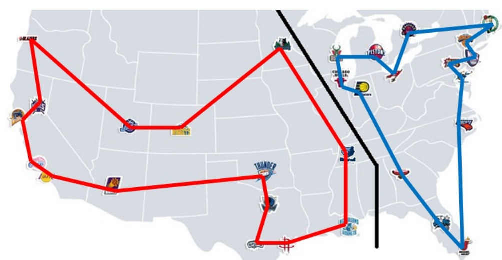

Figure 1: Optimal Hamiltonian Cycles for the NBA teams.

For a graph G on $3k$ vertices, define a P_3-packing to be a set of vertex-disjoint paths of length 3 that cover the vertices of G. In our context, we say that a P_3-packing is *optimal* if the sum of the edge lengths (weights) is minimized. A computer search finds the optimal P_3-packing for graphs G_W and G_E, illustrated in Figure 2 below.

Figure 2: Optimal P_3-packings for the NBA teams.

Given an optimal Hamiltonian cycle of a graph with $3k$ vertices, there are three ways we can remove "every third edge" from this cycle to form a P_3-packing. One of these three P_3-packings is optimal in graph G_W, as we see from the red edges in Figures 1 and 2.

Letting X be the minimum weight of a P_3-packing and Y be the minimum weight of a Hamiltonian cycle, we see that $0 \leq \frac{X}{Y} \leq \frac{2}{3}$, since the sum total of weights of the three trivial P_3-packings is $2Y$, forcing one of the three P_3-packings to have total weight at most $\frac{2Y}{3}$.

Whenever the edge set of the optimal P_3-packing is not a subset of the edge set of the optimal Hamilton cycle (as in graph G_E), the ratio $\frac{X}{Y}$ is strictly less than $\frac{2}{3}$. And often this ratio is much smaller than $\frac{2}{3}$: for example, using the NBA distance matrix (Hoshino and Kawarabayashi 2011), we can compute the values of X and Y to show that $\frac{X}{Y} = \frac{3038}{5690} \sim 0.53$ for the Western Conference and $\frac{X}{Y} = \frac{1765}{3603} \sim 0.49$ for the Eastern Conference.

We now explain how these graph-theoretic ideas can be applied to generate new TTP solutions. We first describe two traditional approaches of matching each team's sequence of three-game road trips with the graph's optimal (or near-optimal) Hamiltonian cycle, generating two canonical schedules frequently used by researchers. We then describe our approach of matching each team's sequence of three-game road trips with the graph's optimal (or near-optimal) P_3-packing, which we then apply to produce best-known solutions for five TTP instances.

Two Standard Canonical Schedules

There is a common three-step heuristic (Rasmussen and Trick 2007) for generating solutions to the n-team TTP. In step 1, we generate double round-robin Home-Away pattern sets (HAPs) in the form of an n by $2(n-1)$ matrix, where the (i, d) entry is H or A depending on whether team t_i is playing a home game or away game on day d. In step 2, we convert these HAPs into timetables which are assignments of matches to time slots. Finally, in step 3, we convert timetables into feasible tournament schedules by assigning each of the n teams a unique row in the matrix.

The ideal timetable is one where each team has many three-game road trips, to reduce the total number of trips required over the course of the $2(n-1)$-day tournament. Such timetables, where the total number of trips is minimal or close-to-minimal, are known as *canonical* schedules, and researchers often use these as initial solutions to difficult n-team TTP instances. After this point (the end of step 2), we can use various heuristics to select one of the $n!$ permutations of $\{t_1, t_2, \ldots, t_n\}$ to create a feasible tournament schedule. Additional techniques, such as simulated annealing and hill-climbing, can then be used to improve these solutions even further, by swapping combinations of rows to minimize the total travel distance.

There are two canonical schedules that have appeared frequently in recent papers, especially those describing TTP approximation algorithms. The first canonical schedule (Westphal and Noparlik 2012) was applied to find a 5.875-approximation algorithm for the generalized version of the TTP, while the second canonical schedule (Yamaguchi et al. 2011) was introduced to find a $\frac{5}{3} + O(\frac{1}{n})$-approximation algorithm for the TTP. As mentioned in the Introduction, Goerigk and Westphal took the first of these canonical schedules as the basis of their analysis for the Galaxy instances, and ran a complex hybrid algorithm to determine best-known TTP solutions (Goerigk and Westphal 2012).

These canonical schedules are effective because in each schedule, every team's sequence of road trips roughly follow the optimal Traveling Salesman tour. To illustrate, suppose the n teams are labelled so that the minimum-weight Hamiltonian cycle is $t_1, t_2, \ldots, t_{n-1}, t_n, t_1$.

A single round-robin tournament schedule can be created using the polygon-circle method (Kirkman 1847); an example of such a schedule for $n = 10$ teams is shown in Table 3. The home-away patterns in this schedule follow a commonly-used home-away assignment scheme (de Werra 1981) that guarantees that no team plays one home (road) game sandwiched between two road (home) games.

	1	2	3	4	5	6	7	8	9
t_1	t_{10}	t_2	t_3	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_6	t_7	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$
t_2	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_{10}	t_3	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_7	t_8
t_3	t_8	t_9	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	$\boldsymbol{t_{10}}$	t_4	t_5	$\boldsymbol{t_6}$	$\boldsymbol{t_7}$
t_4	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_9	t_1	$\boldsymbol{t_2}$	$\boldsymbol{t_3}$	t_{10}	t_5	t_6
t_5	t_6	t_7	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$	t_1	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	t_{10}
t_6	$\boldsymbol{t_5}$	t_{10}	t_7	t_8	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_2	t_3	$\boldsymbol{t_4}$
t_7	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	$\boldsymbol{t_{10}}$	t_8	t_9	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3
t_8	$\boldsymbol{t_3}$	t_4	t_5	$\boldsymbol{t_6}$	$\boldsymbol{t_7}$	$\boldsymbol{t_{10}}$	t_9	t_1	$\boldsymbol{t_2}$
t_9	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	t_5	t_6	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_{10}	t_1
t_{10}	$\boldsymbol{t_1}$	$\boldsymbol{t_6}$	$\boldsymbol{t_2}$	t_7	t_3	t_8	$\boldsymbol{t_4}$	$\boldsymbol{t_9}$	$\boldsymbol{t_5}$

Table 3: A Single Round-Robin Schedule for $n = 10$.

For each $1 \leq i \leq n-1$, let c_i be the slate of matchups and venues for all the games in the i^{th} time slot. In other words, c_i represents the i^{th} column in the schedule above. Let $\overline{c_i}$ denote the exact same matchups as c_i, with the home venues reversed (i.e., if t_j plays a home game against t_k in c_i, then t_j plays a road game against t_k in $\overline{c_i}$).

The first canonical schedule (Westphal and Noparlik 2012) is

$$D_n = c_1, c_2, \ldots, c_{n-2}, c_{n-1}, \overline{c_{n-2}}, \overline{c_{n-1}}, \overline{c_1}, \overline{c_2}, \ldots, \overline{c_{n-3}}.$$

Define $\Gamma_1 = \{c_1, c_2, c_3\}, \Gamma_2 = \{c_4, c_5, c_6\}$, and so on, all the way up to $\Gamma_{(n-1)/3} = \{c_{n-3}, c_{n-2}, c_{n-1}\}$. Then, the second canonical schedule (Yamaguchi et al. 2011) is

$$D^*_n = \Gamma_1, \overline{\Gamma_1}, \overline{\Gamma_2}, \Gamma_2, \Gamma_3, \overline{\Gamma_3}, \ldots, \Gamma_{(n-1)/3}, \overline{\Gamma_{(n-1)/3}}.$$

For example, if $n = 10$, we have

$$D^*_{10} = c_1, c_2, c_3, \overline{c_1}, \overline{c_2}, \ldots, \overline{c_6}, c_4, c_5, \ldots, c_9, \overline{c_7}, \overline{c_8}, \overline{c_9}.$$

This tournament schedule, D^*_{10}, is provided in Table 4.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t_1	t_{10}	t_2	t_3	$\boldsymbol{t_{10}}$	$\boldsymbol{t_2}$	$\boldsymbol{t_3}$	t_4	t_5	$\boldsymbol{t_6}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_6	t_7	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$	$\boldsymbol{t_7}$	t_8	t_9
t_2	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_{10}	t_9	t_1	$\boldsymbol{t_{10}}$	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	t_5	t_3	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_7	t_8	t_6	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$
t_3	t_8	t_9	$\boldsymbol{t_1}$	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$	t_1	t_2	t_{10}	$\boldsymbol{t_4}$	$\boldsymbol{t_2}$	$\boldsymbol{t_{10}}$	t_4	t_5	$\boldsymbol{t_6}$	$\boldsymbol{t_7}$	$\boldsymbol{t_5}$	t_6	t_7
t_4	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_9	t_7	t_8	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_2	t_3	t_1	$\boldsymbol{t_2}$	$\boldsymbol{t_3}$	t_{10}	t_5	t_6	$\boldsymbol{t_{10}}$	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$
t_5	t_6	t_7	$\boldsymbol{t_8}$	$\boldsymbol{t_6}$	$\boldsymbol{t_7}$	t_8	t_9	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	$\boldsymbol{t_9}$	t_1	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	t_{10}	t_3	t_4	$\boldsymbol{t_{10}}$
t_6	$\boldsymbol{t_5}$	t_{10}	t_7	t_5	$\boldsymbol{t_{10}}$	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_9	t_1	t_8	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_2	t_3	$\boldsymbol{t_4}$	$\boldsymbol{t_2}$	$\boldsymbol{t_3}$	t_4
t_7	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	$\boldsymbol{t_4}$	t_5	t_6	t_{10}	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$	$\boldsymbol{t_{10}}$	t_8	t_9	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	t_1	t_2	$\boldsymbol{t_3}$
t_8	$\boldsymbol{t_3}$	t_4	t_5	t_3	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_6	t_7	t_{10}	$\boldsymbol{t_6}$	$\boldsymbol{t_7}$	$\boldsymbol{t_{10}}$	t_9	t_1	$\boldsymbol{t_2}$	$\boldsymbol{t_9}$	$\boldsymbol{t_1}$	t_2
t_9	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_2}$	t_3	t_4	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_7	t_5	t_6	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_{10}	t_1	t_8	$\boldsymbol{t_{10}}$	$\boldsymbol{t_1}$
t_{10}	$\boldsymbol{t_1}$	$\boldsymbol{t_6}$	$\boldsymbol{t_2}$	t_1	t_6	t_2	$\boldsymbol{t_7}$	$\boldsymbol{t_3}$	$\boldsymbol{t_8}$	t_7	t_3	t_8	$\boldsymbol{t_4}$	$\boldsymbol{t_9}$	$\boldsymbol{t_5}$	t_4	t_9	t_5

Table 4: The canonical schedule D^*_{10}.

Note that this definition of D_n^* only holds if $n \equiv 4 \pmod 6$. For the other congruence classes modulo 6, D_n^* is defined in a slightly-different way (Fujiwara et al. 2007).

In the second canonical schedule, this $2(n-1)$-day tournament is split up into blocks of six games, with the first half of each block being the mirror of the second half. We now use a similar idea in creating our tournament schedule, except that each trip's road trips follow an optimal (or near-optimal) P_3-packing.

As mentioned in the Introduction, we now present our three-phase algorithm. In Phase 1, we create a new "canonical" schedule. In Phases 2 and 3, we apply this schedule to determine new upper bounds for various TTP instances.

Phase 1: Schedule based on P_3-packings

Our tournament schedule builds upon a recently-published construction (Hoshino and Kawarabayashi 2012) that establishes a $\frac{4}{3}$-approximation in the special "linear distance" instance where the n teams all lie on a common straight line.

Let m be a positive integer. We first create a single round-robin tournament U on $2m$ teams, and then expand this to a double round-robin tournament Z_n on $n = 2(3m-1)$ teams. Let $\{u_1, u_2, \ldots, u_{2m-1}, x\}$ be the $2m$ teams. Then each team plays $2m-1$ games, according to the polygon-circle Kirkman construction described earlier in Table 3.

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

Table 5: The single round-robin construction for $2m = 8$ teams.

For all games not involving team x, we designate one home team and one road team as follows: for $1 \leq k \leq m$, u_k plays only road games until it meets team x, before finishing the remaining games at home. And for $m+1 \leq k \leq 2m-1$, we have the opposite scenario, where u_k plays only home games until it meets team x, before finishing the remaining games on the road. As an example, Table 5 provides this single round-robin schedule for the case $m = 4$.

This construction ensures that for any match between u_i and u_j, for all $1 \leq i, j \leq 2m-1$, there is exactly one home team and one road team. To verify this, note that u_i is the home team and u_j is the road team iff i occurs before j in the set $\{1, 2m-1, 2, 2m-2, \ldots, m-1, m+1, m\}$.

Now we "expand" this single round-robin tournament U on $2m$ teams to a double round-robin tournament Z_n on $n = 6m-2$ teams. To accomplish this, we transform u_k into three teams, $\{t_{3k-2}, t_{3k-1}, t_{3k}\}$, so that the set of teams in Z_n is $\{t_1, t_2, t_3, \ldots, t_{6m-5}, t_{6m-4}, t_{6m-3}, x\}$.

Suppose u_i is the home team in its game against u_j, played in time slot r. Then we expand that time slot in U into *six* time slots in Z_n, namely the slots $6r-5$ to $6r$. We describe the match assignments in Table 6.

	$6r-5$	$6r-4$	$6r-3$	$6r-2$	$6r-1$	$6r$
t_{3i-2}	$\boldsymbol{t_{3j-1}}$	$\boldsymbol{t_{3j}}$	t_{3j-2}	t_{3j-1}	t_{3j}	$\boldsymbol{t_{3j-2}}$
t_{3i-1}	$\boldsymbol{t_{3j}}$	t_{3j-2}	t_{3j-1}	t_{3j}	$\boldsymbol{t_{3j-2}}$	$\boldsymbol{t_{3j-1}}$
t_{3i}	t_{3j-2}	t_{3j-1}	t_{3j}	$\boldsymbol{t_{3j-2}}$	$\boldsymbol{t_{3j-1}}$	$\boldsymbol{t_{3j}}$
t_{3j-2}	$\boldsymbol{t_{3i}}$	$\boldsymbol{t_{3i-1}}$	$\boldsymbol{t_{3i-2}}$	t_{3i}	t_{3i-1}	t_{3i-2}
t_{3j-1}	t_{3i-2}	$\boldsymbol{t_{3i}}$	$\boldsymbol{t_{3i-1}}$	$\boldsymbol{t_{3i-2}}$	t_{3i}	t_{3i-1}
t_{3j}	t_{3i-1}	t_{3i-2}	$\boldsymbol{t_{3i}}$	$\boldsymbol{t_{3i-1}}$	$\boldsymbol{t_{3i-2}}$	t_{3i}

Table 6: Expanding one time slot in U to six time slots in Z_n.

Recall that by the *each-venue* condition, each team in Z_n must visit every opponent's home stadium exactly once, and by the *at-most-three* condition, road trips are at most three games. We will build a tournament that (nearly) maximizes the number of three-game road trips, and ensure that the majority of these road trips involve three venues closely situated to one another, to minimize total travel. We will also ensure that the *no-repeat* condition is satisfied, so that our final tournament schedule Z_n is feasible.

Group the vertices in the optimal P_3-packing into triplets: $\{t_1, t_2, t_3\}, \{t_4, t_5, t_6\}, \ldots, \{t_{6m-5}, t_{6m-4}, t_{6m-3}\}$. So in Table 6 above, since $\{t_{3j-2}, t_{3j-1}, t_{3j}\}$ are roughly located in the same region, that implies that each of the teams in $\{t_{3i-2}, t_{3i-1}, t_{3i}\}$ can play their three road games against these teams in a highly-efficient manner.

We now explain how to expand the time slots in games involving team x. For each $1 \leq k \leq m$, consider the game between u_k and x. We expand that time slot in U into six time slots in Z_n, as described in Table 7.

	$6r-5$	$6r-4$	$6r-3$	$6r-2$	$6r-1$	$6r$
t_{3k-2}	$\boldsymbol{x}$	$\boldsymbol{t_{3k}}$	t_{3k-1}	x	t_{3k}	$\boldsymbol{t_{3k-1}}$
t_{3k-1}	t_{3k}	$\boldsymbol{x}$	$\boldsymbol{t_{3k-2}}$	$\boldsymbol{t_{3k}}$	x	t_{3k-2}
t_{3k}	$\boldsymbol{t_{3k-1}}$	t_{3k-2}	x	t_{3k-1}	$\boldsymbol{t_{3k-2}}$	$\boldsymbol{x}$
x	t_{3k-2}	t_{3k-1}	$\boldsymbol{t_{3k}}$	$\boldsymbol{t_{3k-2}}$	$\boldsymbol{t_{3k-1}}$	t_{3k}

Table 7: The six time slot expansion for $1 \leq k \leq m$.

And for each $m+1 \leq k \leq 2m-1$, consider the game between u_k and x. We expand that time slot in U into six time slots in Z_n, as described in Table 8.

	$6r-5$	$6r-4$	$6r-3$	$6r-2$	$6r-1$	$6r$
t_{3k-2}	x	$\boldsymbol{t_{3k}}$	$\boldsymbol{t_{3k-1}}$	$\boldsymbol{x}$	t_{3k}	t_{3k-1}
t_{3k-1}	$\boldsymbol{t_{3k}}$	x	t_{3k-2}	t_{3k}	$\boldsymbol{x}$	$\boldsymbol{t_{3k-2}}$
t_{3k}	t_{3k-1}	t_{3k-2}	$\boldsymbol{x}$	$\boldsymbol{t_{3k-1}}$	$\boldsymbol{t_{3k-2}}$	x
x	$\boldsymbol{t_{3k-2}}$	$\boldsymbol{t_{3k-1}}$	t_{3k}	t_{3k-2}	t_{3k-1}	$\boldsymbol{t_{3k}}$

Table 8: The six time slot expansion for $m+1 \leq k \leq 2m-1$.

Let m be fixed. Considering the case $(k, r) = (1, 1)$, we see that teams t_2 and t_3 play each other in time slots 1 and 4. In the final step of our construction, we flip the venues of these two matches so that t_2 is the home team in slot 1 and t_3 is the home team in slot 4. This minor modification creates one fewer trip for t_2.

This construction builds a double round-robin tournament Z_n with $n = 6m - 2$ teams and $2(n-1) = 12m - 6$ time slots, where the three TTP feasibility conditions (*each-venue*, *at-most-three*, *no-repeat*) are all satisfied. To give an example, Table 9 provides the tournament schedule for Z_{10}, corresponding to the case $m = 2$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t_1	$\boldsymbol{t_{10}}$	$\boldsymbol{t_3}$	t_2	t_{10}	t_3	$\boldsymbol{t_2}$	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_4	t_5	t_6	$\boldsymbol{t_4}$	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$	t_7	t_8	t_9	$\boldsymbol{t_7}$
t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_{10}}$	$\boldsymbol{t_1}$	t_3	t_{10}	t_1	$\boldsymbol{t_6}$	t_4	t_5	t_6	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	$\boldsymbol{t_9}$	t_7	t_8	t_9	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$
t_3	t_2	t_1	t_{10}	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	$\boldsymbol{t_{10}}$	t_4	t_5	t_6	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_7	t_8	t_9	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	$\boldsymbol{t_9}$
t_4	$\boldsymbol{t_9}$	$\boldsymbol{t_8}$	$\boldsymbol{t_7}$	t_9	t_8	t_7	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	t_2	t_1	$\boldsymbol{t_{10}}$	$\boldsymbol{t_6}$	t_5	t_{10}	t_6	$\boldsymbol{t_5}$
t_5	t_7	$\boldsymbol{t_9}$	$\boldsymbol{t_8}$	$\boldsymbol{t_7}$	t_9	t_8	t_1	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	t_2	t_6	$\boldsymbol{t_{10}}$	$\boldsymbol{t_4}$	$\boldsymbol{t_6}$	t_{10}	t_4
t_6	t_8	t_7	$\boldsymbol{t_9}$	$\boldsymbol{t_8}$	$\boldsymbol{t_7}$	t_9	t_2	t_1	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	$\boldsymbol{t_5}$	t_4	t_{10}	t_5	$\boldsymbol{t_4}$	$\boldsymbol{t_{10}}$
t_7	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_4	t_5	t_6	$\boldsymbol{t_4}$	t_{10}	$\boldsymbol{t_9}$	$\boldsymbol{t_8}$	$\boldsymbol{t_{10}}$	t_9	t_8	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	t_2	t_1
t_8	$\boldsymbol{t_6}$	t_4	t_5	t_6	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	$\boldsymbol{t_9}$	t_{10}	t_7	t_9	$\boldsymbol{t_{10}}$	$\boldsymbol{t_7}$	t_1	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3	t_2
t_9	t_4	t_5	t_6	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	$\boldsymbol{t_6}$	t_8	t_7	$\boldsymbol{t_{10}}$	$\boldsymbol{t_8}$	$\boldsymbol{t_7}$	t_{10}	t_2	t_1	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{t_1}$	t_3
t_{10}	t_1	t_2	$\boldsymbol{t_3}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	t_3	$\boldsymbol{t_7}$	$\boldsymbol{t_8}$	t_9	t_7	t_8	$\boldsymbol{t_9}$	t_4	t_5	$\boldsymbol{t_6}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$	t_6

Table 9: The schedule Z_{10}, based on a P_3-packing.

Recall that $0 \leq \frac{X}{Y} \leq \frac{2}{3}$, where X is the minimum weight of a P_3-packing and Y is the minimum weight of a Hamiltonian cycle. Intuitively, a smaller value of $\frac{X}{Y}$ indicates that there are many three-team sets clustered together, and so our schedule Z_n should require less travel distance than the standard canonical schedules D_n and D_n^* (see Table 4) based on a minimum-weight Hamiltonian cycle.

To elaborate further, consider an arbitrary TTP instance on $n = 6m - 2$ teams, and let G be the complete graph on the n "vertices" $\{t_1, t_2, \ldots, t_n\}$, where the weight of edge $t_i t_j$ is the value appearing in the i^{th} row and j^{th} column of the corresponding distance matrix. For each $1 \leq i \leq n$, take any polynomial-time algorithm, such as the well-known Christofides method (Christofides 1976), to compute a (near)-optimal Hamiltonian cycle in the graph $G - t_i$. Let Y_i^* be the total edge weight of this Hamiltonian cycle.

For each $1 \leq i \leq n$, take any polynomial-time algorithm to compute a (near)-optimal P_3-packing in the graph $G - t_i$. Now let X_i^* be the minimum total weight among these three P_3-packings. As explained earlier, we have the trivial inequality $0 \leq \frac{X_i^*}{Y_i^*} \leq \frac{2}{3}$ for all values of i. Define $Y := \min(Y_1^*, Y_2^*, \ldots, Y_n^*)$, so that $Y = Y_j^*$ for some index j. Now define $X := X_j^*$.

For example, if we take the 10-team Galaxy instance, we can show that $Y = Y_{10}^* = 238$. From this we determine that $X = X_{10}^* = 119$, with the optimal P_3-packing being $\{t_2, t_1, t_3\}, \{t_6, t_4, t_8\}, \{t_7, t_5, t_9\}$. Thus, we simply reorder the ten teams as $\{t_2, t_1, t_3, t_6, t_4, t_8, t_7, t_5, t_9, t_{10}\}$ to create a schedule Z_{10} that matches the above P_3-packing.

Since $\frac{X}{Y} = \frac{1}{2} < \frac{2}{3}$, we conjecture that Z_{10} requires much less travel distance than the schedule D_{10}. Indeed this is true: the travel distance of Z_{10} is 5164, compared to 5635 for D_{10}. (The results for D_{10}^* are even worse.)

We can go one step further. Because $n = 10$ is such a small case, we can check each of the $n!$ permutations of $\{t_1, t_2, \ldots, t_{10}\}$ to see which row permutation of the tournament schedule gives us the distance-optimal tournament. From this brute-force search, we can show that the optimal solutions of Z_{10} and D_{10} are 4916 and 5559, respectively, which are extremely close to the solutions derived from the optimal P_3-packing and Hamiltonian cycle.

And as our theory predicts, this schedule with distance 4916 has a row permutation that is extremely close to the optimal P_3-packing found above.

Although this brute-force approach is not feasible for large values of n, we now provide a simple heuristic that finds new best bounds for the large Galaxy instances, composed of "teams" based on exoplanets found throughout the galaxy, including Earth. (The pairwise distances represent the number of light years between the exoplanets' host star.) We also apply this heuristic to find close-to-optimal upper bounds for two NFL instances, composed of the teams in the National Football League.

Phase 2: Pairwise-Swapping

Earlier, we described the common three-step heuristic for generating feasible solutions to the n-team TTP. Our tournament schedule, which works for any $n \equiv 4 \pmod 6$ is the end result of the first phase. In Phase 2, we decide how to map the n Galaxy "teams" to $\{t_1, t_2, \ldots, t_n\}$.

As we cannot check all $n!$ permutations to find a *global* optimum, we propose a simple "pairwise-swapping" algorithm that finds a *local* optimum in each iteration. Surprisingly, this elementary heuristic generates tournaments for the Galaxy instances that beat all previously-known solutions for $n \in \{22, 28, 34, 40\}$. In Phase 2, we follow the following four-step sequence:

(a) For each team (vertex) in $\{t_1, t_2, \ldots, t_n\}$, use any polynomial-time algorithm to compute a (near)-optimal P_3-packing of $G - t_k$, with total weight X_k^*. Select the P_3-packing with total weight $X := \min(X_1^*, X_2^*, \ldots, X_n^*)$, and reorder the n teams accordingly. This creates $perm$, a specific permutation of the n teams.

(b) Calculate $dist$, the total travel distance of this schedule.

(c) There are $\binom{n}{2}$ possible choices for the pair (i, j), with $1 \leq i < j \leq n$. List these choices in some *random* order. Starting with the first pair, calculate the total travel distance of the schedule where teams t_i and t_j are swapped. As soon as we find a pair (i, j) for which the resulting schedule has total distance less than $dist$, swap the i^{th} and j^{th} entries of $perm$ and go back to step (b).

(d) End when none of the $\binom{n}{2}$ pairs (i, j) yield a schedule whose total distance is less than $dist$.

We start with the same initial solution (based on our P_3-packing). In each iteration, we swap rows until the algorithm outputs a locally-optimal permutation $perm$ whose total travel distance is $dist$. The decision of which rows are swapped is based on the random ordering described in (c).

For each $n \in \{22, 28, 34, 40\}$ in the Galaxy data set and for each $n \in \{22, 28\}$ in the NFL data set, we apply this "pairwise-swapping" algorithm and iterate until we hit a total running time of 3600 seconds. Our algorithm is coded using Maplesoft, on a stand-alone laptop with 2.75 GB main memory and a single 2.10 GHz processor.

We then choose the tournament that has the lowest travel distance, among all the solutions generated by our iteration. The results are presented in Table 10, comparing our schedules with the best-known results (Trick 2013). In four of the six cases, our tournament schedule is better.

Data Set	Best Known Upper Bound	Output of Phases 1 and 2
Galaxy22	$35,467$	$\mathbf{35,014}$
Galaxy28	$77,090$	$\mathbf{76,518}$
Galaxy34	$146,792$	$\mathbf{145,165}$
Galaxy40	$247,017$	$\mathbf{245,052}$
NFL22	$402,534$	$415,874$
NFL28	$609,788$	$613,574$

Table 10: Table of results for Galaxy and NFL instances.

Phase 3: Hybrid Algorithm

We now take the schedules described in the previous section as starting solutions to a much more powerful heuristic algorithm that combines tabu search with integer programming local search. This is the final phase of our TTP-solving algorithm.

Using this algorithm with weaker canonical schedules as starting solutions already resulted in best-known upper bounds on a large set of instances (Goerigk and Westphal 2012). The basic idea is to iteratively consider the following two neighborhoods:

1. Tabu search step: We begin with improving the starting solutions using a tabu search approach similar to (Di Gaspero and Schaerf 2005). The considered moves in every iteration are the following:
   - Swap home/away pattern for matches between two teams.
   - Swap all matches of two teams.
   - Swap two days.
   - Swap opponents of two teams on a certain day.
   - Swap opponents of a team on two days.

   Additionally, infeasible solutions are considered by using an adaptive, randomized penalty factor on the number of constraint violations.
2. Integer programming step: The resulting solutions are then further improved using an IP solver (in this case, Gurobi v. 5.0). We consider the following two simplified IP formulations:
   - Fix schedule, and optimize the home/away pattern.
   - Fix home/away pattern, and optimize the remaining schedule.

If one step successfully improves the current best solution, the next step begins, until no more improvements can be made. Taking the solutions from Phase 2, we post-improved each of them five times using this approach with randomized parameter choices on a desktop PC with a 2.60 GHz processor. The resulting improvements can be seen in Table 11.

Data Set	Best Known Upper Bound	Output of Phases 1,2,3
Galaxy22	$35,467$	$\mathbf{33,901}$
Galaxy28	$77,090$	$\mathbf{75,276}$
Galaxy34	$146,792$	$\mathbf{143,298}$
Galaxy40	$247,017$	$\mathbf{241,908}$
NFL22	$402,534$	$402,977$
NFL28	$609,788$	$\mathbf{589,123}$

Table 11: Table of improved results for Galaxy and NFL instances.

As we can see from Table 11, we have developed new upper bounds for five instances, each of which beat the previously best-known upper bound by at least 2%. We only reported the results of sets where we found a new upper bound, or came extremely close. For all other instances (e.g. NL16, CIRC16, Super10), our method came close (roughly 1% to 2% worse), but was not an improvement.

Conclusion

Our tournament schedule, based on minimum-weight P_3-packings, generates a feasible solution to the TTP whenever $n \equiv 4 \pmod 6$. The analysis in this paper seems to indicate that our P_3-packing canonical schedule is a better "base" as compared to the standard approach of using the minimum-weight Hamiltonian cycle.

Certainly this was the case in the 10-team Galaxy instance, where the P_3-packing gave an excellent solution which was close to the provably-optimal schedule, and was a much better approximation than either of the canonical schedules found by Hamiltonian cycles.

As demonstrated, our method generalizes to larger n, where we can find a close-to-optimal P_3-packing using any polynomial-time heuristic, and map this permutation of teams (vertices) to find an excellent approximate solution to our TTP instance. We then apply the heuristics described in Phases 2 and 3 to improve this result further, as we showed in beating previously-known upper bounds for five hard benchmark instances.

A natural question is whether there exist similar constructions for $n \equiv 0$ and $n \equiv 2 \pmod 6$. If we can construct canonical schedules in these cases, it is likely that our three-phase approach can generate better solutions to the n-team Galaxy instances, for each $n \in \{24, 26, 30, 32, 36, 38\}$, as well as better solutions to the n-team NFL instances, for each $n \in \{24, 26, 30, 32\}$.

References

Christofides, N. 1976. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, CMU.

de Werra, D. 1981. Scheduling in sports. *Annals of Discrete Mathematics* 11:381–395.

Di Gaspero, L., and Schaerf, A. 2005. A tabu search approach to the traveling tournament problem. In *Proceedings of the 6th Metaheuristics International Conference (MIC-2005).* Available as electronic proceedings.

Easton, K.; Nemhauser, G.; and Trick, M. 2001. The traveling tournament problem: description and benchmarks. *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming* 580–584.

Fujiwara, N.; Imahori, S.; Matsui, T.; and Miyashiro, R. 2007. Constructive algorithms for the constant distance traveling tournament problem. *Lecture Notes in Computer Science* 3867:135–146.

Goerigk, M., and Westphal, S. 2012. A combined local search and integer programming approach to the traveling tournament problem. *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT)* 45–56.

Hentenryck, P. V., and Vergados, Y. 2007. Population-based simulated annealing for traveling tournaments. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 267–271.

Hoshino, R., and Kawarabayashi, K. 2011. The inter-league extension of the traveling tournament problem and its application to sports scheduling. *Proceedings of the 25th AAAI Conference on Artificial Intelligence* 977–984.

Hoshino, R., and Kawarabayashi, K. 2012. The linear distance traveling tournament problem. *Proceedings of the 26th AAAI Conference on Artificial Intelligence* 1770–1778.

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

Kirkman, T. P. 1847. On a problem in combinations. *The Cambridge and Dublin Mathematical Journal* 2:191–204.

Rasmussen, P., and Trick, M. 2007. A Benders approach for the constrained minimum break problem. *European Journal of Operational Research* 177:198–213.

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

Trick, M. 2013. Challenge traveling tournament problems. [Online; accessed 28-June-2013].

Uthus, D.; Riddle, P.; and Guesgen, H. 2012. Solving the traveling tournament problem with iterative-deepening A*. *Journal of Scheduling* 15(5):601–614.

Westphal, S., and Noparlik, K. 2012. A 5.875-approximation for the traveling tournament problem. *Annals of Operations Research* 1–14.

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