
Volume 215, Issue 2, 1 December 2011
ISSN 0377-2217
ELSEVIER
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
Available online at www.sciencedirect.com
ScienceDirect
www.elsevier.com/locate/ejor

European Journal of Operational Research 215 (2011) 481–497

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

A multi-round generalization of the traveling tournament problem and its application to Japanese baseball

Richard Hoshino *, Ken-ichi Kawarabayashi

National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

A R T I C L E   I N F O

Article history:
Received 2 December 2010
Accepted 13 June 2011
Available online 3 August 2011

Keywords:
Scheduling
Timetabling
Graph theory
Perfect matchings
One-factorization
Traveling tournament problem

A B S T R A C T

In a double round-robin tournament involving n teams, every team plays $2(n-1)$ games, with one home game and one away game against each of the other $n-1$ teams. Given a symmetric n by n matrix representing the distances between each pair of home cities, the traveling tournament problem (TTP) seeks to construct an optimal schedule that minimizes the sum total of distances traveled by the n teams as they move from city to city, subject to several natural constraints to ensure balance and fairness. In the TTP, the number of rounds is set at $r = 2$. In this paper, we generalize the TTP to multiple rounds ($r = 2k$, for any $k \geqslant 1$) and present an algorithm that converts the problem to finding the shortest path in a directed graph, enabling us to apply Dijkstra's Algorithm to generate the optimal multi-round schedule. We apply our shortest-path algorithm to optimize the league schedules for Nippon Professional Baseball (NPB) in Japan, where two leagues of $n = 6$ teams play 40 sets of three intra-league games over $r = 8$ rounds. Our optimal schedules for the Pacific and Central Leagues achieve a 25% reduction in total traveling distance compared to the 2010 NPB schedule, implying the potential for considerable savings in terms of time, money, and greenhouse gas emissions.

1. Introduction

There are many practical roles for mathematically-optimal schedules, including supply-chain logistics and airplane flight assignments [9]. Applying combinatorial and graph-theoretic techniques to scheduling optimization helps businesses reduce overall expenditures and travel, leading to economic and environmental benefits. In this paper, we consider scheduling optimization for a professional sports league and develop a theoretical framework to solve a real-life problem.

Globally, major events such as the World Cup and the Olympic Games bring thousands of jobs, urban regeneration, and economic opportunities to host cities [11]. Professional sports leagues such as Major League Baseball, the National Basketball Association, and UEFA European Football involve millions of fans and generate billions of dollars in advertising revenue and television broadcast rights. Despite the appeal and benefit of professional sports leagues, there is a significant economic and environmental impact as teams must travel long distances during the course of a season from one city to another. Finding an optimal schedule that reduces total travel distance is essential, especially in light of today's global economy and uncertain environmental future.

Determining a distance-optimal schedule is known as the *traveling tournament problem* (*TTP*), originally introduced in [7]. In this paper, we present a multi-round generalization of the TTP, and show how it can be solved by reformulating it as a shortest path problem on a digraph. We then apply our scheduling algorithm to the Nippon Professional Baseball (NPB) league in Japan.

The six teams in the NPB Pacific League are spread out all over Japan, including two teams whose home stadiums are separated by a distance of nearly fifteen hundred kilometres. Even in this small island country, during the 2010 season, these six teams traveled a total of 154,390 kilometres to play its 120 intra-league games. In this paper, we produce a provably-optimal schedule that retains all of the constraints of NPB league games that ensure balance and fairness, while reducing the total distance traveled by 40,221 kilometres, representing an improvement of 25.8%. We repeat the analysis for the six teams in the NPB Central League, where the teams are situated more closely. The six teams in the Central League traveled a total of 79,067 kilometres to play its intra-league games during the 2010 season, whereas under our optimal schedule they would only travel 57,836 kilometres. This represents an improvement of 26.8%.

The paper proceeds as follows. In Section 2, we define the traveling tournament problem (TTP), present the known results on the 6-team benchmark set, and describe the primary techniques and methods that have been used in the research. In Section 3, we generalize the TTP to multiple rounds, motivated by the structure of

* Corresponding author.
E-mail address: richard.hoshino@gmail.com (R. Hoshino).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.06.014

Table 1
Optimal TTP solution of NL6, 44 trips, total distance of 23,916 miles.

Team	1	2	3	4	5	6	7	8	9	10
Atlanta	**FLA**	**NYK**	**PIT**	PHI	MON	PIT	**PHI**	**MON**	NYK	FLA
New York	PIT	ATL	FLA	**MON**	**FLA**	PHI	MON	**PIT**	**ATL**	**PHI**
Philadelphia	MON	**FLA**	**MON**	**ATL**	PIT	**NYK**	ATL	FLA	**PIT**	NYK
Montreal	**PHI**	PIT	PHI	NYK	**ATL**	**FLA**	**NYK**	ATL	FLA	**PIT**
Florida	ATL	PHI	**NYK**	**PIT**	NYK	MON	PIT	**PHI**	**MON**	**ATL**
Pittsburgh	**NYK**	**MON**	ATL	FLA	**PHI**	**ATL**	**FLA**	NYK	PHI	MON

Table 2
Distance matrix for the benchmark set NL6 (all distances in miles).

Team	Atlanta	New York	Philadelphia	Montreal	Florida	Pittsburgh
Atlanta	0	745	665	929	605	521
New York	745	0	80	337	1090	315
Philadelphia	665	80	0	380	1020	257
Montreal	929	337	380	0	1380	408
Florida	605	1090	1020	1380	0	1010
Pittsburgh	521	315	257	408	1010	0

professional baseball leagues which involve a small number of teams playing many games against each other during a season (unlike professional soccer and college basketball, where large numbers of teams play just one or two games against each other.) The NPB requires its schedules to be *balanced*, a concept we explain in Section 3. We end the section by formally defining the multi-round balanced traveling tournament problem (mb-TTP). In Section 4, we explain how the mb-TTP can be reformulated as a shortest path problem on a digraph, which enables us to find the optimal multi-round schedule via Dijkstra's Algorithm. In Section 5, we apply the algorithm to the NPB Pacific and Central Leagues, producing the optimal schedules that reduce the total distance traveled by over 25% in both cases. In Sections 6 and 7, we conclude with ideas for implementation, and present directions for future research.

2. The traveling tournament problem

Let there be 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 i and j. By definition, $D_{i,j} = D_{j,i}$ for all $1 \leqslant i, j \leqslant n$, and all diagonal entries $D_{i,i}$ are zero.

In the TTP, a double round-robin schedule is required, where each pair of teams plays twice, once in each other's home venue. The schedule for a TTP is always *compact*, i.e., the tournament lasts $2(n-1)$ days where every team has exactly one game scheduled each day with no byes or days off. The objective is to minimize the total distance traveled by the n teams, where we assume that every team begins the tournament at home and must return home after its last away game.

The standard TTP has two further restrictions: that there be no repeat games on consecutive days (i.e., if teams i and j play on day d, then they cannot play each other again on day $d+1$), and that the length of a home stand or road trip be at most three games. When a team is scheduled for a road trip consisting of multiple away games, the team does not return to their home city but rather proceeds directly to their next away venue. In many ways, the TTP is a variant of the well-known traveling salesman problem (TSP), asking for an 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-complete [17].

Define a *block* to be a feasible solution of the TTP, i.e., a tournament lasting $2(n-1)$ days. We say that a block consists of two *rounds*, with the first round being the first $n-1$ days and the second round being the last $n-1$ days. In our multi-round extension, a tournament will have k blocks, which is equivalent to having $2k$ rounds, or $2k(n-1)$ days. In the TTP, $k = 1$.

The TTP applies *integer programming* to prevent excessive travel, as well as *constraint programming* to create a schedule of home and away games that meet all the feasibility requirements. While each problem is simple to solve on its own, its combination has proven to be extremely challenging [7], even for small cases such as $n = 6$ and $n = 8$.

Given a feasible solution of the TTP, for each team we define a *trip* to be a pair of consecutive games not occurring in the same city (i.e., any situation where that team does not play at home on days d and $d+1$, and therefore has to travel from one venue to another.) For the case $n = 6$, each team plays five home games and five away games. Given that a home stand is at most three games, it is easy to see that each team must make at least seven trips. Thus, in any feasible solution of the TTP, at least $7 \times 6 = 42$ total trips are required by the six teams. It is shown [14] that for the $n = 6$ case, this lower bound is in fact 43. Intuitively, the solution of the TTP that minimizes the objective function of total travel distance will have around 43 trips.

To illustrate with a specific example, Table 1 lists the optimal schedule for a benchmark set [7] known as NL6, consisting of six National League teams from Major League Baseball. The optimal schedule consists of 44 total trips, just one above the fewest possible. In this schedule, as with all subsequent schedules presented in this paper, home games are marked in bold.

In this optimal schedule,[1] the six teams travel a total distance of 23,916 miles, found by applying the distance matrix D in Table 2, which originally appeared in [18].

The TTP has attracted much research activity in recent years, producing new heuristics in integer programming and constraint programming [8], Benders decompositions [3,4], neighbourhood search-based algorithms such as simulated annealing [1], local search approaches such as hill-climbing [13], and a metaheuristic

[1] To illustrate the actual schedule of games in Table 1, consider the first row of the schedule. Atlanta starts off at home, playing three games against Florida, New York, and Pittsburgh. Then Atlanta has three away games (Philadelphia, Montreal, Pittsburgh), before coming home for two games and ending the tournament by playing two away games. Since each team returns to their home venue after their last away game, the total distance traveled by Atlanta is $D_{ATL,PHI} + D_{PHI,MON} + D_{MON,PIT} + D_{PIT,ATL} + D_{ATL,NYK} + D_{NYK,FLA} + D_{FLA,ATL} = 665 + 380 + 408 + 521 + 745 + 1090 + 605 = 4414$. We can repeat the same analysis for each of the other five teams.

based on ejection chains to quickly produce feasible solutions [16]. For additional information, we refer the reader to two comprehensive survey articles on sports scheduling and the TTP [11,15], as well as an up-to-date website with the best-known results on the benchmark sets [18].

In all of these papers, various algorithms have been applied to improve the upper and lower bounds for the n-team TTP. While there are several benchmark sets, the most common ones are based on the teams from MLB's National League. In addition to the six-team NL6 benchmark set, there has been much analysis conducted on NL8, NL10, NL12, NL14, and NL16. We remark that in almost all of these papers, a technique was introduced to improve upon the known bounds rather than to prove optimality, due to the sheer computational complexity of the problem for $n > 6$. The exceptions are two recently-published papers that successfully determined the optimal solution of the 8-team NL8 benchmark set using a novel branch-and-price approach [10] and a depth-first search with upper bounding [19] that stored expensive heuristic estimates in memory to significantly reduce the running time.

For any two teams i and j, we require that in every block they play two games against each other, with one game at each venue. There is no requirement that one match takes place during the first round and the second match takes place during the second round. For example, in the optimal schedule for NL6 presented in Table 1, we see that New York plays Florida on days 3 and 5, i.e., both games between these teams occur in the first round. In Section 3, we will discuss the structure of NPB games and explain how the schedule must be *balanced* with respect to match distribution, i.e., each pair of teams plays exactly once in each round, and if team i hosts team j in round $2t-1$, for some $1 \leqslant t \leqslant k$, then team j must host team i in round $2t$. To see why this condition preserves uniformity and consistency, consider a situation where team x plays most of its games against team i near the beginning of the season, and plays most of its games against team j near the end of the season. If team x experiences substantial roster changes midway through the season (e.g. key players get injured, the team acquires several star players via trades), either team i or team j would benefit tremendously at the expense of the other. Another reason for balance is an economic one: if team i hosts team j several times in the last month of the season, these games will be well-attended by fans, especially if these two teams are natural rivals competing for a playoff position. If team i hosts all of these games, that would provide significant additional revenue to their home city at the expense of team j, not to mention an unfair competitive advantage.

This balancing condition is not part of the TTP, but is alluded to in a constrained version. In [16], the authors add the restriction that the double round-robin schedule be *mirrored*, i.e., if team i hosts team j on day d (where $1 \leqslant d \leqslant n-1$), then team j hosts team i on day $d+n-1$. This variant is known as the mTTP, the mirrored traveling tournament problem, and is a common tournament structure for soccer games in Latin America. Note that the mirrored condition requires the two matches between any two teams to occur $n-1$ days apart, which is stricter than our balancing condition that only requires the two matches take place on non-consecutive days in separate rounds.

Note that any feasible solution of the mTTP is automatically a feasible solution of the TTP, but not conversely. Hence, the optimal solution of the TTP must be no worse than the optimal solution of the corresponding mTTP. To illustrate, Table 3 provides the optimal mTTP solution [16] to the NL6 benchmark set, with 48 trips. This optimal schedule, with total distance of 26,588 miles, has a higher objective value than the optimal solution of the TTP.

3. The multi-round balanced TTP

In this section, we generalize the TTP to include multiple rounds ($r = 2k$, for any $k \geqslant 1$) and describe a natural balancing condition used by Nippon Professional Baseball, and perhaps by other sports leagues around the world. This will motivate us to define the multi-round balanced traveling tournament problem (mb-TTP), for which we will provide an algorithm in Section 4.

While much research has been conducted on the traveling tournament problem, no consideration has been given to tournaments lasting longer than two rounds. Conceivably, this is due to the computational complexity of analyzing a tournament lasting beyond $2(n-1)$ days. However, a multi-round tournament is the correct framework for professional baseball leagues, and so the design of an optimal schedule for NPB must consider this multi-round extension.

A multi-round tournament, with k blocks and $r = 2k$ rounds, consists of $2k(n-1)$ days, where the tournament is partitioned into k blocks of $2(n-1)$ consecutive days. In our generalized version, each of the $\binom{n}{2}$ pairs of matches must occur once each round, with the two games between teams i and j in each block occurring at both venues. This is one criterion of the balancing condition referred to earlier in the paper, and we will adopt this framework as this is how NPB games are structured.

As the context for our paper is baseball, we will now use *sets* rather than *days* to refer to the length of a tournament. Unlike other sports (e.g. football, soccer, hockey, basketball) where a team visits another city to play a single game, baseball leagues always involve a team visiting another city to play multiple games. To avoid any confusion, we will now re-define the TTP (and mb-TTP) to the scheduling of $2k(n-1)$ sets, where each set consists of a fixed number of games played on consecutive days. In the context of the mb-TTP applied to Japanese baseball, $k = 4$ and $n = 6$. Thus, we will create an optimal schedule with $r = 2k = 8$ rounds and $r(n-1) = 40$ sets, with each set consisting of three intra-league games. Hence, the tournament lasts 40 sets, which is equivalent to each team playing 120 intra-league games.

We remark that the multi-round version of the TTP should not be confused with the traveling tournament problem with fixed venues [5], which is a single-round tournament with pre-determined venues (e.g. teams i and j play at the home stadium of team i). In our generalization, we are dealing with multiple rounds where the venues are not fixed; all that is required is that the two matches between teams i and j in rounds $2t-1$ and $2t$ take

Table 3
Optimal mTTP solution of NL6, 48 trips, total distance of 26,588 miles.

Team	1	2	3	4	5	6	7	8	9	10
Atlanta	**MON**	**NYK**	**PHI**	PIT	**FLO**	MON	NYK	PHI	**PIT**	FLO
New York	PHI	ATL	FLO	**MON**	PIT	**PHI**	**ATL**	**FLO**	MON	**PIT**
Philadelphia	**NYK**	PIT	ATL	FLO	**MON**	NYK	**PIT**	**ATL**	**FLO**	MON
Montreal	ATL	FLO	**PIT**	NYK	PHI	**ATL**	**FLO**	PIT	**NYK**	**PHI**
Florida	PIT	**MON**	**NYK**	**PHI**	ATL	**PIT**	MON	NYK	PHI	**ATL**
Pittsburgh	**FLO**	**PHI**	MON	**ATL**	**NYK**	FLO	PHI	**MON**	ATL	NYK

place in different locations (for each $1 \leqslant t \leqslant k$). In each block, whether i hosts j first or second is not pre-determined; this is determined by the algorithm.

In addition to this balancing condition, we will add three more constraints, of which the first two are part of the original TTP framework. In the existing NPB schedule, the maximum length of a home stand or road trip is three sets. Secondly, repeat games do not occur in the NPB, i.e., if team i plays against team j in set s, then these two teams will not play against each other in set $s+1$. Finally, the NPB schedule requires a balance in the number of home and away sets played by each team at any point in the season. More formally, for each ordered pair (i,s) with $1 \leqslant i \leqslant n$ and $1 \leqslant s \leqslant 2k(n-1)$, define $H_{i,s}$ and $R_{i,s}$ to be the number of home and away sets played by team i within the first s sets. By definition, $H_{i,s} + R_{i,s} = s$. In the NPB, we require that $|H_{i,s} - R_{i,s}| \leqslant 2$ for all pairs (i,s). For example, under this requirement a team cannot start or end a season with three consecutive home sets, ensuring that no team gains a momentum-increasing competitive advantage at a key point in the season. We note that this final constraint is not part of the original TTP, as evidenced by the distance-optimal schedule presented in Table 1 where New York opens the season with a three-set road trip and concludes with a three-set home stand.

We now formally define the mb-TTP. As in the TTP, our objective is to minimize the total distance traveled by the n teams, where each team starts and ends the season at home. However, in our formulation, we have k blocks and $r = 2k$ rounds, as well as the following constraints:

(a) The *each-round* condition: Each pair of teams must play exactly once per round, with their matches in rounds $2t-1$ and $2t$ taking place at different venues (for all $1 \leqslant t \leqslant k$).
(b) The *at-most-three* condition: No team may have a home stand or road trip lasting more than three sets.
(c) The *no-repeat* condition: A team cannot play against the same opponent in two consecutive sets.
(d) The *diff-two* condition: $|H_{i,s} - R_{i,s}| \leqslant 2$ for all (i,s) with $1 \leqslant i \leqslant n$ and $1 \leqslant s \leqslant 2k(n-1)$.

The output is a compact schedule consisting of $2k(n-1)$ sets, where each set consists of some fixed number of games played on consecutive days.

To illustrate these conditions, Table 4 provides the optimal balanced schedule for the NL6 benchmark set, and we will develop a rigorous proof of optimality in the following section. By definition, any feasible solution of the mb-TTP (for $k = 1$) is automatically a feasible solution of the TTP. Note that in this optimal schedule, the objective value of 24,684 miles is very close to the best possible value of 23,916 miles for the TTP, and is significantly better than the optimal value of 26,588 miles for the mTTP. Furthermore, this schedule fixes all of the unbalanced properties of the TTP solution for NL6 in Table 1: no longer does a team start/end the season with three consecutive home/away sets, and each pair of teams plays exactly once per round.

In the following section, we present an algorithm for solving the mb-TTP, for any $k \geqslant 1$, by reformulating it as a shortest path problem on a directed graph. We will create a source node and a sink node and link them to numerous vertices in a graph whose (weighted) edges represent the possible blocks that can appear in an optimal schedule. We then apply Dijkstra's Algorithm [6] to find the path of minimum weight between the source and the sink, which is an $O(|V|\log|V| + |E|)$ graph search algorithm that can be applied to any graph or digraph with non-negative edge weights.

4. Shortest path algorithm to solve the mb-TTP

This section will be divided into two subsections. In the first subsection, we explain how the $2k$-round mb-TTP can be reformulated as a shortest path problem on a digraph with an exponential number of vertices. In the second subsection, we describe the four-step construction for the case $n = 6$ to produce a directed graph with $4800k + 2$ vertices and $4{,}104{,}840k - 1{,}481{,}520$ edges, provide the running time for this graph construction, and determine the optimal solution of the mb-TTP for the NL6 benchmark set for each $1 \leqslant k \leqslant 10$.

The mb-TTP can be formulated as an IP problem with $2k(n-1)$ time slots with each slot representing a set. Thus, this IP would have k times as many variables as the original IP formulation for the (NP-complete) double round-robin TTP, and even for the $n = 6$ case, may not be computationally feasible for large k. As a result, we propose a graph-theoretic approach to solving the k-block, $2k$-round mb-TTP by reformulating the optimization problem into a shortest path problem on a digraph. While this methodology is more computationally laborious if k is small, its utility will be demonstrated for large values of k.

4.1. A graph-theoretic reformulation of the mb-TTP

To solve the mb-TTP, we first compute the set of blocks that *can* appear in a distance-optimal tournament. We then introduce a simple "concatenation matrix" to check whether two pre-computed blocks can be joined together to form a multi-block schedule, without violating the *at-most-three* and *no-repeat* conditions. As we will explain, to determine whether blocks B_1 and B_2 can be concatenated, it suffices to check just the last two columns of B_1 and the first two columns of B_2.

By definition, a block is a two-round tournament satisfying the conditions of the mb-TTP, with each of the n teams playing $2(n-1)$ sets of games. Each column of a block represents a set consisting of $\frac{n}{2}$ different *matches*, with each match specifying the two teams as well as the stadium/venue. Thus, a match identifies the home team and away team, not just each team's opponent.

For any column in a block, there are $\binom{n}{n/2}$ ways to select the home teams. Also there are $\binom{n}{n/2} \cdot (\frac{n}{2})!$ ways to specify the matches of any column, since there are $(\frac{n}{2})!$ ways to map any choice

Table 4
Optimal mb-TTP solution of NL6 (for $k = 1$), 44 trips, total distance of 24,684 miles.

Team	1	2	3	4	5	6	7	8	9	10
Atlanta	PHI	**FLO**	**MON**	**NYK**	PIT	MON	NYK	**PIT**	**PHI**	FLO
New York	MON	**PHI**	**PIT**	ATL	FLO	PIT	**ATL**	**FLO**	**MON**	PHI
Philadelphia	**ATL**	NYK	FLO	**PIT**	**MON**	**FLO**	PIT	MON	ATL	**NYK**
Montreal	**NYK**	**PIT**	ATL	FLO	PHI	**ATL**	**FLO**	**PHI**	NYK	PIT
Florida	PIT	ATL	**PHI**	**MON**	**NYK**	PHI	MON	NYK	**PIT**	**ATL**
Pittsburgh	**FLO**	MON	NYK	PHI	**ATL**	**NYK**	**PHI**	ATL	FLO	**MON**

of the $\frac{n}{2}$ home teams to the unselected $\frac{n}{2}$ away teams to decide the set of $\frac{n}{2}$ matches. Hence, there are $m = \binom{n}{n/2}^2 \cdot (\frac{n}{2})!$ different ways we can specify the matches of the first column *and* the home teams of the second column. For $n = 6$, we have $m = \binom{6}{3}^2 \times 3! = 2400$.

There are m ways that the first two columns of a block can be chosen as described above, with the first column listing matches and the second column listing home teams. Now use any method, such as a lexicographic ordering, to index these m options with the integers from 1 to m. By symmetry, there are m different ways we can specify the last two columns of a block, with the last column listing matches and the second-last column listing home teams. Thus, we use the same scheme to index these m options. To avoid confusion, we write the home teams column in binary form, with **1** representing a home game and 0 representing an away game.

For example, (PHI, MON, **ATL**, **NYK**, PIT, **FLO**)^T is one of the 120 possibilities for the matches column, and $(\mathbf{1},\mathbf{1},0,\mathbf{1},0,0)^T$ is one of the 20 possibilities for the home teams column. We remark that if we listed the column of *opponents* rather than the column of matches, there would be only $\frac{120}{2^3} = 15$ unique columns, corresponding to the 15 perfect matchings of the complete graph K_6.

We remark that not all m choices can appear as the index of a (feasible) block. To see this, suppose that the first column is (**MON**, **FLO**, **PIT**, ATL, NYK, PHI)^T and that the second column has home teams $(\mathbf{1},\mathbf{1},\mathbf{1},0,0,0)^T$. Since $|H_{i,3} - R_{i,3}| \leqslant 2$ for all i, no team can start a block with three consecutive home or away sets. It follows that the third column must have home teams $(0,0,0,\mathbf{1},\mathbf{1},\mathbf{1})^T$. This implies that in the first three sets, the teams from Atlanta, New York, and Philadelphia *all* play two home sets followed by an away set. Therefore, no matches can be scheduled between any pair of these teams during the first three sets, implying that each team must play the other two during sets 4 and 5 to satisfy the *each-round* condition. But this is clearly impossible.

For the NL6 benchmark set, there exists some integer p (with $1 \leqslant p \leqslant 2400$) that is the index of the instance where the matches column is (PHI, MON, **ATL**, **NYK**, PIT, **FLO**)^T and the home teams column is $(\mathbf{1},\mathbf{1},0,\mathbf{1},0,0)^T$. Similarly, there exists some q (with $1 \leqslant q \leqslant 2400$) that is the index of the instance where the two columns are (FLO, PHI, **NYK**, PIT, **ATL**, **MON**)^T and $(\mathbf{1},\mathbf{1},0,0,\mathbf{1},0)^T$. The distance-optimal schedule of Table 4 is an example of a block for which the first two columns have index p and the last two columns have index q.

For each pair (u_1, u_2), with $1 \leqslant u_1, u_2 \leqslant m$, define C_{u_2,u_1} to be the $n \times 4$ *concatenation matrix* where the first two columns list the home teams and matches with index u_2, and the next two columns list the matches and home teams with index u_1. For the indices p and q from the previous paragraph, we have

$$C_{q,p} = \begin{bmatrix} \mathbf{1} & \text{FLO} & \text{PHI} & \mathbf{1} \\ \mathbf{1} & \text{PHI} & \text{MON} & \mathbf{1} \\ 0 & \textbf{NYK} & \textbf{ATL} & 0 \\ 0 & \text{PIT} & \textbf{NYK} & \mathbf{1} \\ \mathbf{1} & \textbf{ATL} & \text{PIT} & 0 \\ 0 & \textbf{MON} & \textbf{FLO} & 0 \end{bmatrix}.$$

We now explain the role of m and C_{u_2,u_1} in the construction of our directed graph. Let G consist of a source vertex v_{start}, a sink vertex v_{end}, and vertices $x_{t,u}$ and $y_{t,u}$ defined for each $1 \leqslant t \leqslant k$ and $1 \leqslant u \leqslant m$.

We now describe how these edges are connected, with a pictorial representation of G in Fig. 1. For notational simplicity, denote $v_1 \to v_2$ as the directed edge from v_1 to v_2.

(a) For each $1 \leqslant u \leqslant m$, add the edge $v_{start} \to x_{1,u}$.
(b) For each $1 \leqslant u \leqslant m$, add the edge $y_{k,u} \to v_{end}$.
(c) For each $1 \leqslant t \leqslant k$, and for each $1 \leqslant u_1, u_2 \leqslant m$, add the edge $x_{t,u_1} \to y_{t,u_2}$ iff there exists a (feasible) block for which the first two columns have index u_1 and the last two columns have index u_2.
(d) For each $1 \leqslant t \leqslant k-1$, and for each $1 \leqslant u_1, u_2 \leqslant m$, add the edge $y_{t,u_2} \to x_{t+1,u_1}$ iff the concatenation matrix C_{u_2,u_1} has no row with four home sets, no row with four away sets, and no row with the same opponent appearing in columns 2 and 3.

The following theorem shows that the mb-TTP can be reformulated in a graph-theoretic context, for any $k \geqslant 1$.

Theorem 1. *Every feasible solution of the mb-TTP can be described by a path from v_{start} to v_{end} in graph G. Conversely, any path from v_{start} to v_{end} in G corresponds to a feasible solution of the mb-TTP.*

Proof. By the definition of graph G, any path from v_{start} to v_{end} has length $2k + 1$, and is of the form

$$P = v_{start} \to x_{1,p_1} \to y_{1,q_1} \to x_{2,p_2} \to y_{2,q_2} \to \cdots \to x_{k,p_k} \to y_{k,q_k} \to v_{end}.$$

We first show that P corresponds to a schedule that is a feasible solution of the mb-TTP. For each $1 \leqslant t \leqslant k$, let B_t be any block for which the first two columns have index p_t and the last two columns have index q_t. By part (c) of our construction, since $x_{t,p_t} \to y_{t,q_t}$ is an edge of G, such a block B_t must necessarily exist. Since each B_t is a (feasible) block, all of the balancing constraints of the mb-TTP hold within that block: each pair of teams plays exactly once per round with one match at each venue, no team has a home stand or road trip lasting more than three sets, a team does not play the same opponent in consecutive sets, and $|H_{i,s} - R_{i,s}| \leqslant 2$ for all $1 \leqslant i \leqslant n$ and $1 \leqslant s \leqslant 2(n-1)$.

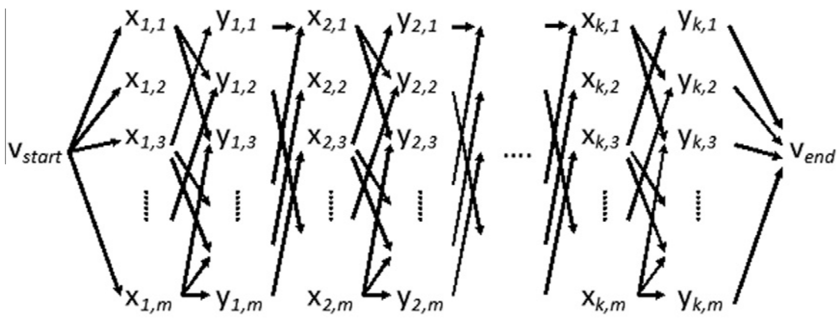

Fig. 1. Reformulation of mb-TTP as a shortest path problem.

We claim that the concatenation of these k blocks, namely $B_1, B_2, \ldots, B_k$, is a feasible solution of the mb-TTP. Clearly each team plays $2k(n-1)$ sets of games, and each pair of teams plays once per round with their matches in rounds $2t-1$ and $2t$ taking place at different venues (for all $1 \leqslant t \leqslant k$). Since each team plays $n-1$ home sets and $n-1$ away sets in every block, we have $H_{i,2t(n-1)} = R_{i,2t(n-1)}$ for all $1 \leqslant t \leqslant k$. This implies that $|H_{i,s} - R_{i,s}| \leqslant 2$ for all $1 \leqslant i \leqslant n$ and $1 \leqslant s \leqslant 2k(n-1)$, since this difference function $|H_{i,s} - R_{i,s}|$ resets to 0 at the end of every block. Therefore, the *each-round* and *diff-two* conditions are satisfied.

To complete our claim, we must justify that the *at-most-three* and *no-repeat* conditions are not violated. Based on the previous paragraph, a violation can only be created in the concatenation of two blocks, specifically in one of the following four situations:

(1) The last column of block B_t matches up with the first column of block B_{t+1} in at least one row, i.e., there exists some team that plays the same opponent in sets $2t(n-1)$ and $2t(n-1)+1$, thereby violating the *no-repeat* condition.
(2) Some row of B_t ends with two home (away) sets and the same row of B_{t+1} begins with two home (away) sets, thereby violating the *at-most-three* condition from set $2t(n-1)-1$ to $2t(n-1)+2$.
(3) Some row of B_t ends with three home (away) sets and the same row of B_{t+1} begins with a home (away) set, thereby violating the *at-most-three* condition from set $2t(n-1)-2$ to $2t(n-1)+1$.
(4) Some row of B_t ends with a home (away) set and the same row of B_{t+1} begins with three home (away) sets, thereby violating the *at-most-three* condition from set $2t(n-1)$ to $2t(n-1)+3$.

Consider the home teams in the second-last column of B_t, the matches in the last column of B_t, the matches in the first column of B_{t+1}, and the home teams in the second column of B_{t+1}. These four columns, representing the sets $2t(n-1)-1$ to $2t(n-1)+2$, are precisely the four columns of the concatenation matrix $C_{q_t,p_{t+1}}$. Since the path P includes the edge $y_{t,q_t} \rightarrow x_{t+1,p_{t+1}}$, the matrix $C_{q_t,p_{t+1}}$ has no row with four home sets, no row with four away sets, and no row with the same opponent appearing in columns 2 and 3. Hence, no team has a four-set home stand or road trip from set $2t(n-1)-1$ to $2t(n-1)+2$, and no team plays the same opponent in sets $2t(n-1)$ and $2t(n-1)+1$. This proves that neither situations (1) or (2) can occur.

As noted before, $H_{i,2t(n-1)} = R_{i,2t(n-1)}$. If row i of B_t ends with three consecutive home or away sets, then $|H_{i,2t(n-1)-3} - R_{i,2t(n-1)-3}| = 3 > 2$, a contradiction. Similarly, if row i of B_{t+1} begins with three consecutive home or away sets, then $|H_{i,2t(n-1)+3} - R_{i,2t(n-1)+3}| = 3 > 2$, a contradiction. This proves that neither situations (3) or (4) can occur.

We have shown that the concatenation of these k blocks, namely $B_1, B_2, \ldots, B_k$, is a feasible solution of the mb-TTP. To conclude the proof, we establish the converse.

Note that any feasible solution of the mb-TTP can be partitioned into k non-overlapping blocks. Let B_t be the tth block of this feasible solution. For this block B_t, let p_t be the index of the first two columns and let q_t be the index of the last two columns. Since B_t is a (feasible) block, it follows that each $x_{t,p_t} \rightarrow y_{t,q_t}$ is an edge in G. Since pairwise blocks can be concatenated without violating the constraints of the mb-TTP, $y_{t,q_t} \rightarrow x_{t+1,p_{t+1}}$ in an edge in G for all $1 \leqslant t \leqslant k-1$. We conclude that $P = v_{start} \rightarrow x_{1,p_1} \rightarrow y_{1,q_1} \rightarrow x_{2,p_2} \rightarrow y_{2,q_2} \rightarrow \cdots \rightarrow x_{k,p_k} \rightarrow y_{k,q}k \rightarrow v_{end}$ is a path connecting v_{start} to v_{end}. □

Having constructed our digraph, we now assign a *weight* to each edge using the distance matrix D so that the shortest path (i.e., path of minimum total weight) from v_{start} to v_{end} corresponds to the optimal solution of the mb-TTP that minimizes the total distance traveled by the n teams.

For any block, we define its *in-distance* to be the total distance traveled by the n teams within that block, i.e., starting from set 1 and ending at set $2(n-1)$. Note that the in-distance does not include the distance traveled by the teams heading *to* the venue of set 1 or *from* the venue of set $2(n-1)$. We will use this definition in part (c) below.

To illustrate, consider the distance-optimal one-block schedule presented in Table 4, with total distance 24684. The in-distance of this block is 21579. Since each team begins the season at home and returns home after having played their last away game, three teams travel $D_{ATL,PHI} + D_{NYK,MON} + D_{FLO,PIT} = 2012$ miles to play their first set and three teams travel $D_{FLO,ATL} + D_{PHI,NYK} + D_{PIT,MON} = 1093$ to return home after their final set. The total traveling distance of this two-round tournament is therefore $2012 + 21579 + 1093 = 24684$.

Having defined the in-distance of a block, we are now ready to assign edge weights.

(a) For each $1 \leqslant u \leqslant m$, the weight of edge $v_{start} \rightarrow x_{1,u}$ is the total distance traveled by the $\frac{n}{2}$ teams making the trip from their home city to the venue of their opponent in set 1.
(b) For each $1 \leqslant u \leqslant m$, the weight of edge $y_{k,u} \rightarrow v_{end}$ is the total distance traveled by the $\frac{n}{2}$ teams making the trip from the venue of their opponent in set $2k(n-1)$ back to their home city.
(c) For each $1 \leqslant t \leqslant k$, and for each $1 \leqslant u_1, u_2 \leqslant m$, the weight of edge $x_{t,u_1} \rightarrow y_{t,u_2}$ is the *minimum* in-distance of a block, selected among all blocks for which the first two columns have index u_1 and the last two columns have index u_2.
(d) For each $1 \leqslant t \leqslant k-1$, and for each $1 \leqslant u_1, u_2 \leqslant m$, the weight of edge $y_{t,u_2} \rightarrow x_{t+1,u_1}$ is the total distance traveled by the teams that travel from their match in set $2t(n-1)$ to their match in set $2t(n-1)+1$, where the last two columns of the tth block have index u_2 and the first two columns of the $(t+1)$th block have index u_1.

To illustrate (d), consider the two-block schedule produced by concatenating two copies of Table 4. Then this is a feasible solution of the mb-TTP for $k = 2$, with path $P = v_{start} \rightarrow x_{1,p} \rightarrow y_{1,q} \rightarrow x_{2,p} \rightarrow y_{2,q} \rightarrow v_{end}$, having total weight $2012 + 21579 + 2818 + 21579 + 1093 = 49081$. The distance 2818, representing the weight of edge $y_{1,q} \rightarrow x_{2,p}$, is the distance traveled by the teams from their matches in set 10 to their matches in set 11. Specifically, this total distance is $D_{FLO,PHI} + D_{PHI,MON} + D_{PIT,MON} + D_{FLO,PIT}$, the distances traveled by Atlanta, New York, Montreal, and Florida, respectively.

By this construction, we have produced a weighted digraph. In part (c), suppose there exist two blocks B and B' for which the first two columns have index u_1 and the last two columns have index u_2. If the in-distance of B is less than the in-distance of B', then block B' cannot be a block in an optimal solution, since we can just replace B' by B to create a feasible solution with a lower objective value. This trivial observation, based on Bellman's Principle of Optimality, allows us to assign the *minimum* in-distance as the weight of edge $x_{t,u_1} \rightarrow y_{t,u_2}$, for all $1 \leqslant u_1, u_2 \leqslant m$. As a result, we have a digraph G on $2mk + 2$ vertices and at most $2m + (2k-1)m^2$ edges, with a unique weight for each edge. Combined with the previous theorem, we have established the following.

Theorem 2. *Let $P = v_{start} \rightarrow x_{1,p_1} \rightarrow y_{1,q_1} \rightarrow x_{2,p_2} \rightarrow y_{2,q_2} \rightarrow \cdots \rightarrow x_{k,p_k} \rightarrow y_{k,q_k} \rightarrow v_{end}$ be a shortest path in G from v_{start} to v_{end}, i.e., a path that minimizes the total weight. For each $1 \leqslant t \leqslant k$, let B_t be the block of minimum in-distance selected among all blocks for which the first two columns have index p_t and the last two columns have index*

q_t. Then the multi-block schedule $S = B_1, B_2, \ldots, B_k$, created by concatenating the k blocks consecutively, is an optimal solution of the mb-TTP.

Therefore, we have shown that the mb-TTP is isomorphic to a finding the shortest weighted path in the directed graph G. We now build G for the $n = 6$ case, and to illustrate the construction, we build G for the NL6 benchmark set, applying the distance matrix in Table 2. In Section 5, we will repeat this construction twice more, applying the distance matrices for the Pacific and Central Leagues.

4.2. Construction of the digraph for n = 6

For $n = 6$ and a fixed k, the directed graph G consists of $2mk + 2 = 4800k + 2$ vertices. From the four-step construction described in the previous subsection, we can determine the set of edges, and assign the appropriate weights to each edge. Steps (a) and (b) clearly add $m = 2400$ edges to G. In step (c), we construct edge $x_{t,u_1} \to y_{t,u_2}$ for each $1 \leqslant t \leqslant k$ iff there is a (feasible) block for which the first two columns have index u_1 and the last two columns have index u_2. In step (d), we construct edge $y_{t,u_2} \to x_{t+1,u_1}$ for each $1 \leqslant t \leqslant k-1$ iff the concatenation matrix C_{u_2,u_1} has no row with four home sets, no row with four away sets, and no row with the same opponent appearing in columns 2 and 3.

In Appendix A, we show that of the $m^2 = 5{,}760{,}000$ choices for the ordered pair (u_1, u_2), exactly 2,618,520 produce an edge of G in step (c) and 1,486,320 produce an edge of G in step (d). As a result, G is a directed graph with $4800k + 2$ vertices and $2400 + 2400 + 2{,}618{,}520k + 1{,}486{,}320(k-1) = 4{,}104{,}840k - 1{,}481{,}520$ edges. Therefore, incrementing k by one adds over four million edges to graph G. We now apply Dijkstra's shortest path algorithm to obtain the k-block $2k$-round schedule that minimizes the total travel distance.

Let M and N be 2400×2400 matrices that store the weights of the edges constructed in steps (c) and (d). Specifically, for each $1 \leqslant u_1,\ u_2 \leqslant 2400$ and any t, $M[u_1,u_2]$ is the weight of edge $x_{t,u_1} \to y_{t,u_2}$, and $N[u_1,u_2]$ is the weight of edge $y_{t,u_2} \to x_{t+1,u_1}$. If $x_{t,u_1} \to y_{t,u_2}$ is not an edge in G (i.e., there is no block for which the first two columns have index u_1 and the last two columns have index u_2), we set $M[u_1,u_2] := c$, where c is a massive number that dominates the weights of all actual edges. Similarly, if $y_{t,u_2} \to x_{t+1,u_1}$ is not an edge in G (i.e., the concatenation matrix C_{u_2,u_1} does not satisfy the required conditions), we set $N[u_1,u_2] := c$. This way, all the information on the edges and edge weights can be stored in these two matrices M and N, while assuring that a non-edge will not appear in the solution to Dijkstra's Algorithm.

Dijkstra's Algorithm constructs a shortest path tree from the initial vertex to every other vertex in the graph. For each vertex v in G, let $w(v)$ be the weight of the shortest path from v_{start} to that vertex. By definition, $w(v_{start}) = 0$. For each $1 \leqslant u \leqslant 2400$, $w(x_{1,u})$ is the total distance traveled by the $\frac{n}{2}$ teams making the trip from their home city to the venue of their opponent in set 1. For each $1 \leqslant u \leqslant m$, we have the following:

$$w(y_{t,u}) = \min_{1 \leqslant u^* \leqslant m} \{w(x_{t,u^*}) + M[u^*,u]\}, \quad \text{for } 1 \leqslant t \leqslant k,$$
$$w(x_{t+1,u}) = \min_{1 \leqslant u^* \leqslant m} \{w(y_{t,u^*}) + N[u^*,u]\}, \quad \text{for } 1 \leqslant t \leqslant k-1.$$

The last step of this recursive process generates $w(y_{k,u})$, the weight of the shortest path from v_{start} to each $y_{k,u}$. For each u, define $last(u)$ to be the total distance traveled by the $\frac{n}{2}$ teams making the trip from the venue of their opponent in set $2k(n-1)$ back to their home city. By symmetry, it is clear that $last(u) = w(x_{1,u})$. Therefore, we have

$$w(v_{end}) = \min_{1 \leqslant u^* \leqslant m} \{w(y_{k,u^*}) + w(x_{1,u^*})\}.$$

The value of $w(v_{end})$ is the distance of the optimal solution to the k-block mb-TTP, corresponding to the path

$$P = v_{start} \to x_{1,p_1} \to y_{1,q_1} \to x_{2,p_2} \to y_{2,q_2} \to \cdots \to x_{k,p_k} \to y_{k,q_k} \to v_{end}.$$

From this optimal path, we can immediately generate the optimal schedule, as it is just the concatenation of the t blocks $B[p_t,q_t]$, for $t = 1,2,3,\ldots,k$. It is straightforward to see that for any t, the shortest path from v_{start} to $x_{t,u}$ must have the same total weight as the shortest path from $y_{k+1-t,u}$ to v_{end}. To determine the value of $w(v_{end})$, we will exploit this symmetry to reduce the computational time by one-half.

We now provide the running time of this graph construction and apply Theorem 2 to determine the optimal solution of the mb-TTP for the NL6 benchmark set. All of the code was written in Maple and compiled using Maplesoft 13 using a single Toshiba laptop under Windows with a single 2.10 Gigahertz processor and 2.75 Gbytes RAM.

It takes Maple less than 0.1 seconds to complete step (a) of the digraph construction, assigning a weight to each edge $v_{start} \to x_{1,u}$. Step (b), which assigns a weight to each edge $y_{k,u} \to v_{end}$, also takes 0.1 seconds. Step (d), which determines the weights of all edges $y_{1,u_2} \to x_{2,u_1}$, and stores this information in the 2400×2400 matrix N, takes just 53 seconds. Finally, step (c) takes almost 5 hours (17,346 seconds) of computation time. This is by far the most expensive calculation in the algorithm.

Finally we apply Dijkstra's Algorithm. For each k, applying the above recursive procedure to find the shortest path from v_{start} to v_{end} takes approximately $25k$ seconds. Hence, incrementing k by one adds about 25 seconds to the total computation time. Once we have determined the matrices M and N, Dijkstra's Algorithm enables us to determine the optimal 100-round schedule ($k = 50$) in just 20 minutes. While our proposed graph-theoretic approach may not be computationally efficient for small values of k, its utility is clearly evident for large values of k. This is summarized in Table 5.

For the $k = 1$ case, the optimal solution has total distance 24,684, corresponding to the schedule provided in Table 4. The optimal solution for the mb-TTP for the four-block, eight-round ($k = 4$) mb-TTP is presented in Table 6. To save space, each team has been denoted by a single letter. To avoid ambiguity, Philadelphia has been marked as (H) and Pittsburgh as (P).

The total distance of this tournament is 95,341 miles. The corresponding path in G is described by

$$P = v_{start} \to x_{1,p_1} \to y_{1,q_1} \to x_{2,p_2} \to y_{2,q_2} \to x_{3,p_3} \to y_{3,q_3} \to x_{4,p_4} \to y_{4,q_4} \to v_{end},$$

Table 5
Running time for each component of the mb-TTP algorithm.

Process	Edge Construction	Edges Created (#)	Time (s)
Step (a)	$v_{start} \to x_{1,u}$	2400	0.1
Step (b)	$y_{k,u} \to v_{end}$	2400	0.1
Step (c)	$x_{t,u_1} \to y_{t,u_2}$	2,618,520	17,346
Step (d)	$y_{t,u_2} \to x_{t+1,u_1}$	1,486,320	53
Dijkstra for $k = 1$	None	None	32
Dijkstra for $k = 2$	None	None	55
Dijkstra for $k = 3$	None	None	79
Dijkstra for $k = 4$	None	None	102
Dijkstra for $k = 5$	None	None	128
Dijkstra for $k = 6$	None	None	141
Dijkstra for $k = 7$	None	None	174
Dijkstra for $k = 8$	None	None	189
Dijkstra for $k = 9$	None	None	221
Dijkstra for $k = 10$	None	None	245

with the edge weights 1093, 21579, 2818, 21579, 1082, 22544, 1974, 21579, 1093, respectively. In this optimal schedule, we have $p_1 = p_2$ and $q_1 = q_2$, which explains why the first two blocks in Table 6 are identical. We remark that the 40 columns of Table 6 can be written backwards to produce another optimal schedule with the same objective value of 95341. By symmetry, there must be at least two optimal schedules, each being the counterpart of the other.

5. Application to Japanese baseball

Nippon Professional Baseball (NPB) consists of the six-team Pacific League and the six-team Central League. All teams play 144 games during a season, with 120 intra-league games and 24 inter-league games. Specifically, an NPB team plays twelve home games and twelve away games against each of the other five teams in its league (24×5) in addition to two home games and two away games against all six teams in the other league (4×6). All twenty-four inter-league games take place during a common five-week stretch beginning in mid-May, right near the start of the season.

Our algorithm for the mb-TTP will only apply to the 120 intra-league games; the remaining 24 inter-league games will not be examined in this paper as it is a separate optimization problem based on the theory of bipartite tournaments [2]. Since the large majority of NPB games are intra-league, we have decided to focus our optimization efforts there. For the purposes of this paper, we will assume that the NPB schedule consists of 120 intra-league games, ignoring the 24 inter-league games and the annual July break for the all-star game. When we demonstrate that the Pacific and Central League schedules can be improved by over 25% with respect to total travel distance, we are strictly referring to the scheduling of these 120 intra-league games.

As in Major League Baseball, nearly all NPB games occur in sets of three games. Thus, we will adopt the same structure when building our schedule. We will assume that the $n = 6$ teams in each league plays 40 sets of three games. Hence, we require a compact schedule with $k = 4$ blocks and $r = 2k = 8$ rounds, producing $r(n - 1) = 40$ sets of matches per team. We note that a compact schedule is the ideal way to build an optimized schedule, as not having three matches on a given day stretches out the season, which is especially problematic when multiple rainouts on successive days require numerous games to be rescheduled late in the season.

In our compact schedule, the 120 games can be played in just 20 weeks (assuming six games per week, with one day off) rather than the existing non-compact schedule which requires over 22 weeks to complete, not factoring in the time to complete rescheduled games due to rain. Furthermore, a compact schedule makes it more likely for all teams to end their regular season on the same day, heightening the drama for teams competing for playoff spots.

Notwithstanding the practicality of a compact schedule, we note that 40 sets of three games is the natural way to build an optimized schedule, instead of the existing NPB schedule which has several exceptions. For example, one team (Chiba Marines) in the Pacific League plays three sets of two away games against another team (Orix Buffaloes) rather than two sets of three away games. This arrangement requires Chiba to make an unnecessary additional 582 kilometre trip to play Orix. Hence, we will build a compact schedule with 40 sets, solving the mb-TTP for the case $n = 6$ and $r = 8$.

In this section, we complete the full analysis for the Pacific League, and in Appendix C, we repeat the same analysis for the Central League. As illustrated in Fig. 2, the six teams in the NPB Pacific League are spread out all over Japan, with two teams (Fukuoka and Hokkaido) separated by a distance of nearly 1500 kilometres.

Depending on the distance between cities, teams travel by bus, taxi, shinkansen bullet-train, or airplane. Often the stadium is far away from the train station or airport, and so an additional bus ride is required to arrive at the stadium. These factors have all been taken into account (see Appendix B) to produce Table 7, the 6×6 distance matrix for the Pacific League.

We remark that several NPB teams schedule their home games outside of their regular stadiums, to bring professional baseball to more rural areas of Japan. Since these games occur rarely, and these rural areas are often quite close to the actual home stadium, this has not been considered in our analysis. The only exception is the Kansai-based Orix Buffaloes, who play half of their home games in Osaka and the other half in nearby Kobe located 50 kilometres away. As a result, we have created a bogus venue called "Orix Stadium" located halfway between the two home stadiums and used that to compute the values in our distance matrix.

Having calculated our distance matrices, we are now ready to apply the algorithm developed in Section 4 to determine the distance-optimal schedules for the NPB Pacific League. We show how our optimal schedule achieves a reduction of over 25% in total travel distance compared to the initial schedule for the 2010 NPB season, ignoring the additional games that needed to be rescheduled due to rainouts. Applying our mb-TTP shortest path algorithm, we determine an optimal schedule for the eight rounds of the NPB intra-league season. This schedule is presented in Table 8.

Fig. 2. Location of the six Pacific League teams in Japan.

Table 6
Optimal solution of the mb-TTP ($k = 4$) for the NL6 benchmark set.

Team	R1	R2	R3	R4	R5	R6	R7	R8
Atlanta (A)	F**HP**NM	P**NMF**H	F**HP**NM	P**NMF**H	PF**MNH**	MNH**PF**	H**FMN**P	MN**PH**F
New York (N)	H**MFA**P	FA**PH**M	H**MFA**P	FA**PH**M	**HP**FAM	**FAM**HP	M**HP**AF	P**AFM**H
Philadelphia (H)	**N**AMP**F**	**MP**FN**A**	**N**AMP**F**	**MP**FN**A**	N**MP**FA	P**FAN**M	**A**NF**PM**	**F**PMA**N**
Montreal (M)	PN**HFA**	HFA**PN**	PN**HFA**	HFA**PN**	**F**HAP**N**	**AP**NF**H**	**NP**AFH	**AFH**NP
Florida (F)	**AP**NMH	**NMH**AP	**AP**NMH	**NMH**AP	M**ANH**P	NH**PM**A	PA**HMN**	HMN**PA**
Pittsburgh (P)	**M**FA**HN**	**A**HNM**F**	**M**FA**HN**	**A**HNM**F**	**A**NH**MF**	**H**MFA**N**	**F**MNH**A**	**NH**AF**M**

Table 7
Distance matrix for the Pacific League (all distances in kilometres).

Team	Chiba	Tohoku	Hokkaido	Orix	Fukuoka	Saitama
Chiba	0	361	904	582	934	86
Tohoku	361	0	580	670	1100	374
Hokkaido	904	580	0	1115	1466	928
Orix	582	670	1115	0	595	595
Fukuoka	934	1100	1466	595	0	958
Saitama	86	374	928	595	958	0

In addition to comparing the total distance traveled by each team, we also enumerate the number of trips taken between each of the $\binom{6}{2} = 15$ pairs of cities. We present the 6 by 15 matrix representing this information, with a separate row for each team. For example, "ME" denotes the number of trips taken between the home stadiums of the Chiba Marines (M) and the Tohoku Eagles (E). Under the 2010 NPB schedule, every team made at least 32 trips, not counting additional trips to play rescheduled rainout games. Under our optimal schedule, no team would make any more than 29 trips. This information is provided in Tables 9 and 10.

Finally, we compare these two schedules with respect to the total distance traveled by all six teams, using our distance matrix. Overall, our optimal schedule achieves a 25.8% reduction in total travel compared to the existing schedule, in addition to a 18.8% reduction in total trips taken. Table 11 lists the improvement on a team by team basis.

In addition to the significant 25.8% reduction in total distance traveled, we also remark that this is a more equitable schedule. In the 2010 NPB schedule, one team (Fukuoka Hawks) traveled nearly 12,500 kilometres more than another team (Saitama Lions). Under the proposed optimal schedule, the difference between the most traveled and least traveled would reduce to just 4500 kilometres.

Finally, we measure the *schedule efficiency*, a simple metric that evaluates the quality of an existing schedule relative to the best possible and worst possible schedules with respect to total distance traveled. Specifically, we define this metric as

$$\textit{Schedule Efficiency} = \frac{\textit{DistWorst-DistExisting}}{\textit{DistWorst-DistOptimal}},$$

where *DistWorst*, *DistOptimal*, and *DistExisting* are the total distances traveled under the worst, optimal, and existing schedules, respectively. By this formula, the worst possible schedule has efficiency 0% while the optimal schedule has efficiency 100%.

We know the values of *DistOptimal* and *DistExisting* from the above analysis. To calculate *DistWorst*, we modify our Maple code to output the longest path rather than the shortest path, ensuring that each of the 2,618,520 blocks in step (c) of our graph construction have the highest in-distance among all blocks with indices u_1 and u_2. We find that for the Central League, *DistWorst* equals 177,502. From this, we determine that the efficiency of the 2010 NPB schedule is just 37%.

We repeat the same algorithm for the Central League, producing a distance-optimal schedule that achieves a 26.8% reduction in total distance traveled and a 14.6% reduction in total trips taken, compared to the 2010 NPB schedule. Under the existing schedule, which has an efficiency of 57%, one team (Hiroshima Carp) traveled nearly 7500 kilometres more than another team (Tokyo Swallows). Under the proposed schedule, the difference between the most

Table 8
Optimal schedule for the Pacific League, total distance of 114,169 kilometres.

Team	R1	R2	R3	R4	R5	R6	R7	R8
Chiba Marines (M)	**L**FE**HB**	**E**HBL**F**	**E**HBL**F**	**BL**FE**H**	**F**BH**LE**	**H**LEF**B**	E**LF**BH	F**BHE**L
Tohoku Eagles (E)	F**HM**BL	M**BLF**H	MF**LHB**	LHB**MF**	**LFBH**M	BH**ML**F	**MF**BHL	**BHL**MF
Hokkaido Fighters (F)	**EM**BLH	**BLH**EM	L**EHB**M	HB**ML**E	ME**LBH**	LBH**ME**	**H**EML**B**	**ML**BHE
Orix Buffaloes (B)	H**LFE**M	FE**MH**L	H**LM**FE	M**FEH**L	**HM**EFL	**EFL**HM	L**HEM**F	EM**FL**H
Fukuoka Hawks (H)	**B**ELM**F**	**LM**FB**E**	**BM**FEL	**FEL**BM	B**LM**EF	M**EFB**L	FB**LEM**	LEM**FB**
Saitama Lions (L)	MB**HFE**	HFE**MB**	**FB**E**MH**	**E**MHF**B**	**E**HFM**B**	**FM**BE**H**	**B**MH**FE**	**H**FEB**M**

Table 9
Trips taken between Pacific League venues under the existing schedule.

	ME	MF	MB	MH	ML	EF	EB	EH	EL	FB	FH	FL	BH	BL	HL	Total
Chiba Marines (M)	5	7	6	5	5	0	1	2	0	0	0	1	2	1	1	36
Tohoku Eagles (E)	5	1	1	0	1	7	6	6	6	0	0	2	1	0	1	37
Hokkaido Fighters (F)	0	6	0	1	1	5	1	0	2	5	5	3	1	1	1	32
Orix Buffaloes (B)	0	2	6	1	1	1	6	0	1	4	1	0	5	5	1	34
Fukuoka Hawks (H)	1	1	0	6	0	0	1	4	2	1	7	1	6	0	5	35
Saitama Lions (L)	0	1	1	1	5	0	0	0	8	1	0	6	2	4	5	34
Total Trips	11	18	14	14	13	13	15	12	19	11	13	13	17	11	14	208

Table 10
Trips taken between Pacific League venues under our optimal schedule.

	ME	MF	MB	MH	ML	EF	EB	EH	EL	FB	FH	FL	BH	BL	HL	Total
Chiba Marines (M)	4	4	2	3	5	3	0	0	1	0	1	0	4	2	0	29
Tohoku Eagles (E)	3	2	1	1	1	6	3	2	4	0	0	0	3	1	2	29
Hokkaido Fighters (F)	4	1	0	1	2	4	0	0	0	3	3	3	3	2	1	27
Orix Buffaloes (B)	2	1	3	1	1	4	2	0	0	2	0	1	6	5	1	29
Fukuoka Hawks (H)	1	1	2	3	1	2	0	2	3	2	2	1	4	0	3	27
Saitama Lions (L)	0	1	1	2	4	2	3	0	3	0	3	2	0	4	3	28
Total Trips	14	10	9	11	14	21	8	4	11	7	9	7	20	14	10	169

Table 11
Comparison of existing and optimal schedules for the Pacific League.

	Distance (existing)	Distance (optimal)	Reduction in distance (%)	Trips (existing)	Trips (optimal)	Reduction in trips (%)
Chiba Marines	23,266	16,606	28.6	36	29	19.4
Tohoku Eagles	23,710	17,975	24.2	37	29	21.6
Hokkaido Fighters	28,599	20,234	29.2	32	27	15.6
Orix Buffaloes	24,128	18,713	22.4	34	29	14.7
Fukuoka Hawks	33,352	21,143	36.6	35	27	22.9
Saitama Lions	20,885	19,498	6.6	34	28	17.6
Total	153,940	114,169	25.8	208	169	18.8

traveled and least traveled would reduce to just 4000 kilometres. The complete details appear in Appendix C.

6. Implementation

Naturally, there are additional factors involved with the actual scheduling of NPB games at these home stadiums. For example, one of the ballparks hosts a three-day concert each August, and another ballpark is used as the locale of the national high school baseball tournament. Hence those teams must play away games on those particular days. In many sports leagues, rival teams play "derby matches" [12] that need to be scheduled on particular days to optimize revenue, and are often dictated by the wishes of television broadcasters. Sometimes a league has two outstanding teams that are bitter rivals (e.g. the Boston Red Sox and the New York Yankess in MLB), hence officials will deliberately schedule a match between them to conclude the season, to add drama and boost TV ratings. These constraints must be taken into account when producing an optimal schedule that can be implemented by NPB, to ensure no conflicts occur, and that the schedule is best possible for all parties involved.

Note that our optimal schedule is *compact*, and lasts 20 weeks with six games for each team each week. This seems far preferable to the 2010 NPB schedule where many days had only a handful of scheduled games, adding two extra weeks to the regular season, and causing the teams to finish their season on different dates. Perhaps there is a good reason for why this was the case; if so, this will need to be incorporated into our analysis.

We also remark that for the NPB, we must deal with the optimal method of scheduling the few "home" games that certain teams play outside of their home stadium. While most of these special games take place in nearby cities, there is one notable exception. The Hokkaido Fighters in the Pacific League play nine home games (three sets) at the Tokyo Dome, home of the Yomiuri Giants in the Central League. Hence, our optimal schedule can be improved even further by ensuring that these special home games taking place in Tokyo occur alongside a road trip to Saitama and/or Chiba, two teams whose home stadiums are just a short bus ride away. Of course, we must add the constraint that these special games do not coincide with actual home games of the Yomiuri Giants.

Furthermore, the NPB may consider relaxing the *diff-two* condition that requires $|H_{i,s} - R_{i,s}|$ to be at most 2 for all pairs (i,s). Specifically, if the NPB would like to develop a schedule that is even better than the distance-optimal schedule produced by our algorithm, this condition could be changed to $|H_{i,s} - R_{i,s}| \leqslant 3$. By changing the *diff-two* condition to *diff-three*, the set of feasible timetables would significantly increase, thereby creating more blocks in our search space. Therefore, the optimal NPB schedule for the Pacific League would be even better than the schedule presented in Table 8, resulting in an overall reduction of more than 25.8%. However, for reasons described in Section 7, our shortest-path algorithm may no longer be computationally feasible as our $n \times 4$ concatenation matrix would need to be expanded by two columns.

Finally, we remark that our analysis has been conducted using a 6×6 *distance* matrix, given the context of the traveling tournament problem which seeks to minimize the total distance traveled. However, if NPB officials wished to develop a schedule that minimized the total *cost* of team travel, then the distance matrix could be replaced by a cost matrix and the same algorithm repeated to produce an optimal cost-effective schedule.

7. Conclusion

We have developed a rigorous algorithm to solve the multi-round balanced traveling tournament problem (mb-TTP) for the case $n = 6$ and applied it to develop optimal intra-league schedules for the two six-team leagues in Nippon Professional Baseball, showing how both leagues can reduce their total traveling distance by over 25%. However, our algorithm is unlikely to be computationally feasible for $n \geqslant 8$. Even for the $n = 8$ case, we have $m = \binom{n}{n/2}^2 \cdot (\frac{n}{2})! = 117{,}600$, requiring us to store the edge weights of digraph G in two $117{,}600 \times 117{,}600$ matrices. Hence, we require more sophisticated techniques for solving the mb-TTP for $n \geqslant 8$. Since NL8 was recently solved using a depth-first-search with upper bounding [19] as well as a branch-and-price approach [10], perhaps these powerful methods could be applied in conjunction with our Dijkstra formulation to solve the mb-TTP for $n = 8$.

Also a natural question is to extend the standard TTP to multiple rounds, eliminating the *each-round* and *diff-two* conditions of our balanced framework. However, this problem may be quite difficult as the *each-round* condition enabled us to apply the theory of perfect matchings and one-factorizations to enumerate our feasible timetables, and the *diff-two* condition allowed us to reduce our concatenation matrix to just four columns. Without the *diff-two* condition, we would require six columns rather than four, since we need to check cases when one team ends a block with three consecutive home (away) games or starts the next block with three consecutive home (away) games, to validate the *at-most-three* condition. To see why this is so, we refer the reader to the Proof of Theorem 1, where situations (3) and (4) were presented. Therefore, to apply the same shortest path algorithm for the multi-round standard TTP would require us to choose $m = \binom{n}{n/2}^3 \cdot (\frac{n}{2})!$, which requires $m = 48{,}000$ for $n = 6$. Once again, this approach is not likely to be computationally feasible given the size of the $m \times m$ matrices.

Our mb-TTP algorithm outputs the schedule that minimizes the total distance traveled by the n teams, as that is the objective function. However, perhaps a better objective function would be to minimize the gap between the most traveled team and the least traveled team, to optimize competitive balance. In our analysis of the Pacific League, this gap of 12,500 kilometres in the existing schedule can be reduced to just 4500 kilometres under our schedule. But surely this gap can be reduced even further by changing the objective function, producing a schedule where every team travels essentially the same distance during the course of a season.

Finally, one can analyze the *carry-over effect* [11] to produce fair schedules. Team i is said to give a carry-over effect to team j if some other team x plays against i on day d and against j on day $d+1$. If all n teams are equally strong, then the carry-over effect is irrelevant; however, if team i is extremely weak (or strong) and many teams play team j immediately after playing team i, then team j may have a competitive advantage (or disadvantage). Let C denote the carry-over matrix of a schedule, where the entry $C_{i,j}$ represents the number of times team i gives a carry-over to team j. In an ideal schedule, C would be constant. As an illustration, Tables 12 and 13 provide the carry-over matrices for the existing Pacific League schedule as well as for our distance-optimal schedule. We see that neither is well-balanced with respect to carry-over effects.

One tantalizing research problem would be to create a multi-round schedule that achieves the best balance between three competing variables: the total distance traveled by the n teams, the gap between the most traveled and least traveled, and the difference between the largest and smallest entries in the carry-over matrix C. By creating an objective function that combines all three of these variables, we would produce a tournament schedule that is fair and balanced to all teams, while simultaneously addressing economic concerns and environmental impact.

Table 12
Carry-over matrix for the Pacific League, for the existing schedule.

Team	Marines	Eagles	Fighters	Buffaloes	Hawks	Lions
Marines	0	7	11	7	6	10
Eagles	7	0	7	12	12	3
Fighters	13	6	0	11	3	8
Buffaloes	4	8	7	0	8	13
Hawks	10	9	8	8	0	5
Lions	7	11	8	2	11	0

Table 13
Carry-over matrix for the Pacific League, for the distance-optimal schedule.

Team	Marines	Eagles	Fighters	Buffaloes	Hawks	Lions
Marines	0	5	13	9	5	7
Eagles	14	0	5	2	12	6
Fighters	4	14	0	13	2	6
Buffaloes	8	7	2	0	11	11
Hawks	7	6	8	9	0	9
Lions	6	7	11	6	9	0

Appendix A

In this appendix, we describe the full process of creating the digraph G for the case $n = 6$. To do this, we follow the four-step procedure given in Section 4.1.

Step (a) of the construction is straightforward as for each $1 \leqslant u \leqslant m$, we only need to look at the first column listing the matches, and can ignore the home teams listing in the second column. There are $\frac{n}{2} = 3$ matches in the first set, of which all the away teams will travel to their opponent's home venue to start the season. Hence, each edge weight $v_{start} \to x_{1,u}$ is calculated by adding $\frac{n}{2}$ elements of the distance matrix D, as we did in Table 4 to determine that the weight of $v_{start} \to x_{1,p}$ is $D_{ATL,PHI} + D_{NYK,MON} + D_{FLO,PIT} = 2012$.

Note that this sum is a simple function of how the six teams are grouped into three pairs, and is not dependent on which three teams play at home. Thus, there are only 15 possible values for the edge weight $v_{start} \to x_{1,u}$, corresponding to the 15 perfect matchings of the complete graph K_6. Hence, each of these 15 sums occurs $\binom{6}{3} \cdot 2^3 = 160$ times among the 2400 edge weights of $v_{start} \to x_{1,u}$. By symmetry, step (b) of the construction works the exact same way and is equally straightforward.

Now we proceed with steps (c) and (d). First, we note that these two constructions are independent of t. Thus, to construct the edges $x_{t,u_1} \to y_{t,u_2}$, it suffices to determine all the edges from x_{1,u_1} to y_{1,u_2} with the corresponding weights and replicate that k times. To construct the edges $y_{t,u_2} \to x_{t+1,u_1}$, it suffices to determine all the edges from y_{1,u_2} to y_{2,u_1} with the corresponding weights and replicate that $k-1$ times. Regardless of k, we only need to compute steps (c) and (d) once, and we will store all of the necessary information in $m \times m$ matrices, which for the $n = 6$ case would be matrices with dimensions 2400×2400.

To perform step (c) of the construction, we must ensure that the weight of each edge $x_{1,u_1} \to y_{1,u_2}$ is the *minimum* in-distance of a block for which the first two columns have index u_1 and the last two columns have index u_2. To accomplish this, we adopt the three-phase approach of [14] that is a common heuristic for solving the two-round TTP. Phase one generates double round-robin home-away pattern (HAP) sets in the form of an n by $2(n-1)$ matrix, phase two converts these HAP sets into timetables which are assignments of matches to time slots, and phase three converts timetables into feasible schedules (i.e., blocks) by assigning each team of NL6 a unique row in the matrix. Once we have enumerated all possible blocks, we will be able to assign the proper weight to each edge $x_{1,u_1} \to y_{1,u_2}$ by applying the distance matrix D. To illustrate, Tables 14 and 15 provide a HAP set with 44 total trips satisfying the mb-TTP conditions, with a corresponding timetable.

To turn a timetable into a block, one simply maps the six teams of NL6 to any of the 6! permutations of $\{1,2,3,4,5,6\}$. We note that any timetable can be turned into a block, but a HAP set does not necessarily produce a feasible timetable. For example, consider a HAP set with two identical rows, say in rows i and j. Then this HAP set cannot correspond to a feasible timetable as teams i and j cannot play one another as there is no time slot where one team is home while the other is on the road.

We now generate all possible HAP sets and from this, determine the set of all possible timetables. Of course, this process will create numerous schedules which almost certainly would not appear as a block of an optimal solution, regardless of the distance matrix D. But for the sake of rigour, we perform the entire enumeration. Later in Section 4, we discuss ways to avoid searching the entire search space and reduce the computation time from hours to seconds. While phases one and two are long and

Table 14
A balanced HAP set with 44 total trips.

Team	1	2	3	4	5	6	7	8	9	10	Trips
1	**H**	A	A	**H**	**H**	**H**	A	A	A	**H**	7
2	A	**H**	**H**	**H**	A	A	A	**H**	**H**	A	8
3	A	A	**H**	**H**	**H**	A	A	A	**H**	**H**	7
4	**H**	**H**	A	A	A	**H**	**H**	**H**	A	A	7
5	**H**	A	A	A	**H**	**H**	**H**	A	A	**H**	7
6	A	**H**	**H**	A	A	A	**H**	**H**	**H**	A	8

Table 15
A timetable corresponding to the balanced 44-trip HAP set in the above table.

Team	1	2	3	4	5	6	7	8	9	10
1	**2**	6	3	**5**	**4**	**3**	5	4	2	**6**
2	1	**3**	**4**	**6**	5	4	6	**5**	**1**	3
3	5	2	**1**	**4**	**6**	1	4	6	**5**	**2**
4	**6**	**5**	2	3	1	**2**	**3**	**1**	6	5
5	**3**	4	6	1	**2**	**6**	**1**	2	3	**4**
6	4	**1**	**5**	2	3	5	**2**	**3**	**4**	1

detailed, they are necessary in order to complete step (c) of our digraph construction.

First we determine all possible HAPs that satisfy the constraints of the mb-TTP. By the *diff-two* condition, each team must play at least two home sets and two away sets in the first (five-set) round, as $|H_{i,5} - R_{i,5}| \leqslant 2$ for each $1 \leqslant i \leqslant 6$. As each feasible timetable generates 6! blocks by mapping each permutation of the six rows to the six teams, we may impose a fixed structure on the HAP sets knowing that all feasible blocks will be enumerated. So without loss of generality, assume that $H_{i,5} = 3$ for $1 \leqslant i \leqslant 3$ and $H_{i,5} = 2$ for $4 \leqslant i \leqslant 6$. Let us represent this one-round scenario by the vector $(3,3,3,2,2,2)^T$.

In each column, there are $\binom{6}{3} = 20$ ways to designate the home teams. Hence, there are $20^4 = 160{,}000$ different ways to select the first four columns. We can immediately reject any combination for which a single row consists of all home sets or all away sets as that violates the *at-most-three* condition. Now consider all combinations for which $H_{i,4} = 1$ for some $1 \leqslant i \leqslant 3$ or $H_{i,4} = 3$ for some $4 \leqslant i \leqslant 6$. We can reject all of these cases as the desired $(3,3,3,2,2,2)^T$ pattern is then impossible to achieve. For all other combinations, there is a unique way of selecting the three home teams in the fifth column to achieve the $(3,3,3,2,2,2)^T$ pattern.

We may eliminate any HAP set with two identical rows, as this violates the *each-round* condition that stipulates that every pair of teams must play once in the first round. Considering all 160,000 combinations for these first five columns, and discarding all the illegal cases, we find that only 10,800 possibilities remain. One such example is the first round (i.e., first five columns) of Table 14.

Of these 10,800 possibilities, some rows contain the pattern (**H**, **H**, **H**, A, A) or (A, A, A, **H**, **H**) which violates the *diff-two* condition that $|H_{i,3} - R_{i,3}| \leqslant 2$. After eliminating these combinations as well, we are finally left with 5940 ways we can assign a HAP set to the first round to create the $(3,3,3,2,2,2)^T$ pattern, where $H_{i,5} = 3$ for $1 \leqslant i \leqslant 3$ and $H_{i,5} = 2$ for $4 \leqslant i \leqslant 6$.

By the *each-round* condition, each team must play five home sets and five away sets in a block. By symmetry, there are 5940 ways we can assign a HAP set to the second round to create the $(2,2,2,3,3,3)^T$ pattern, i.e., so that $H_{i,10} - H_{i,5} = 2$ for $1 \leqslant i \leqslant 3$ and $H_{i,10} - H_{i,5} = 3$ for $4 \leqslant i \leqslant 6$. Considering all 5940×5940 concatenations of the two rounds into a single 10-column HAP set, we note that there are 3! ways to permute the first three rows and 3! ways to permute the last three rows, implying that each HAP set's six rows appears together 36 times. Hence, we can reduce the search space by a factor of 36, by ensuring that no HAP set in the search space can be obtained from any other HAP set via a row permutation.

Of the $\frac{5940}{36} \times 5940 = 980{,}100$ potential HAP sets for a 10-set block, we discard all sets violating the *at-most-three* condition, which occur anytime one-round rows such as (A, A, **H**, **H**, **H**) and (**H**, A, A, **H**, A) are concatenated. There are 627,944 options that remain, any of which can be a HAP set for a feasible timetable. We group these 627,944 possibilities according to the number of *trips* required to play this double round-robin tournament, assuming each team starts and ends the season at home. This is displayed in Table 16.

Ignoring row permutations, there is only one HAP set with 44 trips, which is provided in Table 14. Intuitively, we reason that in a distance-optimal schedule, each block will contain around 44 trips. We will find that in our optimal multi-round schedules for the NPB Pacific and Central Leagues, *every* block contains 44 trips with a HAP set that is a row permutation of Table 14. While we may wish to ignore HAP sets with a large number of trips (say 50 or more), at this point we will not do that to ensure that the output of our shortest path algorithm is guaranteed to be optimal. There may exist a strange distance matrix D for which the optimal solution contains a block with many trips, and for this reason we will consider all 627,944 possible HAP sets in our analysis.

For each possible HAP set, we now determine all possible timetables that fit that pattern, as we did with our example in Tables 14 and 15. Since each round consists of fifteen distinct matches, with three matches played over five sets, this is equivalent to a *one-factorization* of the complete graph K_6, i.e., a partition of the edges of K_6 into five perfect matchings [20]. From the first half of the timetable in Table 15, we see that a possible one-factorization of K_6 is

$$[\{12,35,46\},\{16,23,45\},\{13,24,56\},\{15,26,34\},\{14,25,36\}].$$

For each $1 \leqslant i \leqslant 5$, the ith element in the above one-factorization is a perfect matching of K_6 corresponding to the three matches played in set i of Table 15.

There are six different one-factorizations of K_6, each of which can be permuted in 5! different ways. Therefore, one round of a HAP set can be mapped column by column to a one-factorization of K_6 in $6 \times 5! = 720$ different ways. However, virtually none of these 720 possibilities produce a feasible timetable due to a contradiction in at least one of the fifteen pairings where both teams are playing at home (or on the road) in that column. For example, the first half of the HAP set in Table 14 yields a feasible timetable in only 8 out of 720 cases, including the one listed in the first five columns of Table 15.

For each of the 627,944 feasible HAP sets, we consider all 720×720 possible (two-round) timetables that can be derived, and eliminate the blocks that are not feasible. To do this, we check the *each-round* condition to ensure that team i plays team j twice, with one match at each venue, and also verify the *no-repeat* condition to ensure that no team plays the same opponent in sets 5 and 6. This produces 169,728 feasible timetables. We group these 169,728 possibilities according to the number of trips required to play this double round-robin tournament, assuming each team starts and ends the season at home. The results are listed in Table 17.

Of the $627{,}944 \times 720 \times 720$ possible timetables, only 169,728 are feasible. In the large majority of the cases, a perfect matching does not properly align with the home-away assignment for a given column, thereby creating the contradiction described earlier. For the remaining cases, note that the *each-round* condition further stipulates that if i hosts j in the first round, then j must host i in the second round. If this condition is not met in any one of the fifteen cases (i.e., there exists a pair (i,j) for which i hosts j in both rounds), then the timetable is not feasible. That explains why so many potential timetables are discarded, leaving only 169,728 that are feasible. Each feasible timetable yields 6! different blocks that are candidates to appear in an optimal solution of the mb-TTP.

There are $6! \times 6! \times 2^{15}$ blocks satisfying the *each-round* condition, since there are $6 \times 5! = 6!$ ways of selecting the first five columns, 6! ways of selecting the last five columns, and 2^{15} ways of selecting the home teams for the 15 matches in the first round. However, the above analysis shows that there are only $169{,}728 \times 6!$ schedules that also satisfy the *diff-two*, *no-repeat*, and *at-most-three* conditions. Since $\frac{6! \times 6! \times 2^{15}}{169728 \times 6!} \sim 139$, we have reduced the search space by a factor of 139, compared to a brute-force enumeration of all possible blocks.

Having completed our three-phase approach, we are finally ready to complete step (c), determining the set of edges $x_{1,u_1} \rightarrow y_{1,u_2}$ that appear in G, along with the weight of each edge. To do this, we create a 2400×2400 matrix M with all entries initially set to 10^5 and a 2400×2400 matrix B with all entries initially blank.

We let T_i be the ith timetable and p_j be the jth permutation of the six teams in NL6. For each $1 \leqslant i \leqslant 169{,}728$ and $1 \leqslant j \leqslant 720$, define $S_{i,j}$ to be the 10-set schedule created by replacing the numbers

Table 16
Enumeration of the possible HAP sets for a 10-set block, grouped by number of trips.

Trips	<44	44	45	46	47	48	49	50	51	52	53	54	55	56	>56
HAP sets	0	1	84	1224	9130	41,056	107,140	166,897	161,812	97,296	35,268	7360	632	44	0

Table 17
Enumeration of the possible timetables for a 10-set block, grouped by number of trips.

Trips	<44	44	45	46	47	48	49	50	51	52	53	54	55	56	>56
Timetables	0	8	48	820	3592	16,076	36,384	49,376	38,704	18,228	5464	948	80	0	0

Table 18
Optimal solution of the mb-TTP for each $k \leqslant 10$, with restrictions on the search space.

Trips	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$	Timetables (#)	Time (s)
$\leqslant$44	24,684	47,598	70,944	95,341	119,738	143,973	167,420	191,817	216,214	240,449	8	0.3
$\leqslant$45	24,684	47,598	70,944	95,341	119,610	143,819	167,340	191,737	216,006	240,215	56	3.4
$\leqslant$46	24,684	47,598	70,944	95,341	119,403	143,687	167,134	191,531	215,518	239,877	876	59
All	24,684	47,598	70,944	95,341	119,403	143,687	167,134	191,531	215,518	239,877	169,728	17,346

$\{1,2,3,4,5,6\}$ in T_i with the six teams from NL6 according to the mapping $\{1,2,3,4,5,6\} \to p_j$. Note that each of these 169,728 × 720 possibilities results in a (feasible) block, of which one example appears in Table 4. Now run a double for-loop, performing the following three-part procedure over all $S_{i,j}$, for each i from 1 to 169,728 and each j from 1 to 720.

(i) If $S_{i,j}$ is a block for which Atlanta plays two home sets in the first round, then stop and do not proceed to part (ii).
(ii) Otherwise, let u_1 and u_2 be the indices of the first two columns and the last two columns of $S_{i,j}$. Apply the distance matrix D to calculate the in-distance of $S_{i,j}$. Let this value be $d(S_{i,j})$. If $d(S_{i,j}) \geqslant M[u_1,u_2]$ then stop and do not proceed to part (iii).
(iii) Otherwise, set $M[u_1,u_2] := d(S_{i,j})$, $M[u_2,u_1] := d(S_{i,j})$, $B[u_1,u_2] := S_{i,j}$, and $B[u_2,u_1] := S'_{i,j}$, where $S'_{i,j}$ is the block $S_{i,j}$ written backwards, so that the cth column of $B[u_2,u_1]$ is identical to the $(11-c)$th column of $B[u_1,u_2]$, for all $1 \leqslant c \leqslant 10$.

For any iteration, if $S_{i,j}$ reaches step (iii), then $S_{i,j}$ is a block for which Atlanta plays three home sets in the first round, with $d(S_{i,j})$ less than the value of $M[u_1,u_2]$ at that point in the double for-loop. Whenever this occurs, we replace the entries $M[u_1,u_2]$, $M[u_2,u_1]$, $B[u_1,u_2]$, and $B[u_2,u_1]$ as described above. We claim the following:

Theorem 3. *For each $1 \leqslant u_1, u_2 \leqslant 2400$, $x_{1,u_1} \to y_{1,u_2}$ is an edge of G iff $M[u_1,u_2] < 10^5$ after the completion of the double for-loop. Furthermore, $B[u_1,u_2]$ is the block of minimum in-distance, selected among all blocks for which the first two columns have index u_1 and the last two columns have index u_2. Finally, the in-distance of this optimal block is $M[u_1,u_2]$.*

Proof. For each of the 169,728 × 6! possible blocks, we set Atlanta as a pivot row, following the symmetry-reduction technique described in [19]. Each block B has a unique block counterpart B' by writing its columns in reverse order. Since each team plays five home sets in a block, each team must play two or three home sets in a round to satisfy the *diff-two* condition. If Atlanta plays two (three) home sets in the first round of block B, then Atlanta plays three (two) home sets in the first round of block B'. Since the in-distance of B is clearly equal to the in-distance of B', we may disregard half of the cases by symmetry. Thus, in part (i) above, we may ignore all blocks for which Atlanta plays two home sets in the first round, thereby reducing the computation time by a factor of two. Note that by symmetry, $x_{1,u_1} \to y_{1,u_2}$ is an edge of G iff $x_{1,u_2} \to y_{1,u_1}$ is an edge of G. Furthermore, the weights of these two edges must be equal.

Hence, we only consider blocks $S_{i,j}$ for which Atlanta plays three home sets in the first round. Let u_1 and u_2 be the indices of the first two columns and the last two columns of $S_{i,j}$. Then by the above three-part procedure that cycles through every possible choice for i and j, $M[u_1,u_2]$ is guaranteed to be the minimum in-distance among all blocks for which the first two columns have index u_1 and the last two columns have index u_2. For each ordered pair (u_1,u_2), when this minimum in-distance is achieved for some i and j, the entry $B[u_1,u_2]$ is set to the block $S_{i,j}$. Hence, when the procedure terminates after having considered all 169,728 × 720 cases, $B[u_1,u_2]$ is guaranteed to be the block of minimum in-distance. If $B[u_1,u_2]$ remains null, then there does not exist a block for which the first two columns have index u_1 and the last two columns have index u_2.

To conclude the proof, we must explain the choice of the entry 10^5 in initializing the matrix M. By Table 17, every $S_{i,j}$ has fewer than 56 trips, and so the in-distance of $S_{i,j}$ sums up the distance of fewer than $56 - 3 - 3 = 50$ trips, not counting the trips taken by the three teams traveling to their opponents' venue in set 1 and the three teams traveling from their opponents' venue in set 10. Hence, $d(S_{i,j}) < 50z$, where z is the maximum value that appears in the distance matrix D. Thus, when we initialize the matrix M, we just need to set each entry to a value higher than $50z$. Since $z = 1380$ for the NL6 case (see Table 2), $10^5 > 50z = 69{,}000$ is permissible. Hence, $x_{1,u_1} \to y_{1,u_2}$ is an edge in G iff $M[u_1,u_2] < 10^5$ after the completion of the double for-loop. □

There are $m^2 = 5{,}760{,}000$ possible edges $x_{1,u_1} \to y_{1,u_2}$. Running this double for-loop, we discover that 2,618,520 (about 45%) actually appear as edges in G. For the purposes of this shortest path algorithm, a non-edge can be replaced by a fake edge with a massive weight (such as 10^5) that dominates the weights of all actual edges. By doing this, all the information on edges and edge weights can be stored in this single matrix M. Of course, when we apply Dijkstra's Algorithm to find the path that minimizes the total weight, a fake edge of weight 10^5 will not appear in the output.

Finally we proceed with step (d) of our construction, which will determine the edges $y_{1,u_2} \to x_{2,u_1}$ in G, along with their edge weights. By definition, $y_{1,u_2} \to x_{2,u_1}$ is an edge in G iff the concatenation matrix C_{u_2,u_1} has no row with four home sets, no row with four away sets, and no row with the same opponent appearing in columns 2 and 3. Let N be a 2400×2400 matrix with all entries initially set to 0.

Create a 120×120 matrix R and a 400×400 matrix T. Note that there are $120 = \binom{6}{3} \times 3!$ ways to select the matches of any one column, and $400 = \binom{6}{3} \times \binom{6}{3}$ ways to select the home teams of any two columns. Index the 120 possibilities for the matches column from 1 to 120, and do likewise with the 400 possibilities for the two columns of home teams. For each pair (u_2, u_1), with $1 \leqslant u_2, u_1 \leqslant 2400$, determine the indices $q, p \in [1,120]$ corresponding to columns 2 and 3 of the concatenation matrix C_{u_2,u_1} as well as the indices $s, r \in [1,400]$ representing the home teams that appear in the four columns of C_{u_2,u_1}.

For each $1 \leqslant p,q \leqslant 120$, consider the middle two columns of the concatenation matrix C_{u_2,u_1}, and let $L[q,p]$ be the distance traveled by all the teams that make a trip from their match in column 2 to their match in column 3. Now let R be the matrix that assigns a penalty value if these two columns violate the *no-repeat* condition, where some team plays the same opponent in consecutive sets. Specifically, define $R[q,p] := 10^5$ if the *no-repeat condition* is violated, and $R[q,p] := 0$ otherwise. By symmetry, $R[p,q] = R[q,p]$ for all choices of p and q.

For each $1 \leqslant r, s \leqslant 400$, consider the home teams appearing in the four columns of the concatenation matrix C_{u_2,u_1}. Let T be the matrix that assigns a penalty value if these four columns violate the *at-most-three* condition, where some team plays four consecutive home sets or four consecutive away sets. Specifically, define $T[s,r] := 10^5$ if the *at-most-three condition* is violated, and $T[s,r] := 0$ otherwise. By symmetry, $T[r,s] = T[s,r]$ for all choices of r and s.

Now define $N[u_2,u_1] := L[q,p] + R[q,p] + T[s,r]$. By definition, $N[u_2,u_1] \geqslant 10^5$ if either of the above two balancing conditions are violated. If neither condition is violated, then $R[q,p] = 0$ and $T[s,r] = 0$, implying that $N[u_2,u_1] = L[q,p]$. As in the proof of Theorem 3, let z be the maximum value appearing in the distance matrix D. Since $L[q,p] \leqslant 6z < 10^5$, we have established the following theorem.

Theorem 4. *For each $1 \leqslant u_1, u_2 \leqslant 2400$, $y_{1,u_2} \to x_{2,u_1}$ is an edge of G iff $N[u_2,u_1] < 10^5$. Furthermore, the weight of this edge is $N[u_2,u_1]$.*

By symmetry, $N[u_1,u_2] = N[u_2,u_1]$ for all $1 \leqslant u_1, u_2 \leqslant 2400$. Of the m^2 = 5,760,000 possible edges that can appear, we find that 1,486,320 (about 25%) actually appear as edges in G. As explained earlier, we can replace non-edges with fake edges having a weight greater than 10^5, knowing that the output of our shortest path algorithm will not include these non-edges.

Maple required 13 hours to generate the 627,944 feasible HAP sets, and required an additional 67 hours of computation time to turn these HAP sets into the 169,728 feasible timetables. While this process takes over three days of computation time, this procedure needs to be only run once, and then the 169,728 timetables can be stored in a file and applied to any distance matrix D. For the interested reader, the authors are happy to provide the files containing these 169,728 timetables, grouped by the number of trips in each timetable, as well as all of the Maple code used in the production of this paper.

The weakness of the algorithm occurs in part (c) of our graph construction, where the enumeration of all $169{,}728 \times 6!$ possible blocks takes nearly five hours of computation time. Suppose we do not consider all 169,728 timetables, but only those with a small number of total trips. From Table 17, we know that at least 44 trips are required, including the six taken by the teams to get to the venue of set 1 and from the venue of set 10. Suppose that instead of considering all timetables, we only considered those which had 46 or fewer trips, conjecturing that those with 47 or more trips cannot (or are unlikely to) appear as a block in an optimal schedule. Under this assumption, the computation time is reduced significantly. By Table 17, there are only 8 + 48 + 820 = 876 timetables with at most 46 trips. Since $\frac{169728}{876} \sim 193$, we can reduce computation time by a factor of nearly 200. Although this smaller subset no longer guarantees an optimal solution to the mb-TTP algorithm, perhaps the solution to the simplified problem will indeed be optimal for a large percentage of distance matrices D.

Table 18 lists the optimal value of the NL6 mb-TTP for each $1 \leqslant k \leqslant 10$, for each of four cases: when we restrict the search space to timetables with at most 44 trips, 45 trips, 46 trips, and 55 trips. Note that the last case encompasses all 169,728 timetables, as all feasible timetables have at most 55 trips by Table 17.

From the Table 18, we see that the optimal solution to the k-round NL6 mb-TTP is always a concatenation of blocks with at most 46 trips, for each $k \leqslant 10$. We have confirmed this to be the case for each $k \leqslant 50$ and conjecture that it is indeed true for every k. As a result, we can ignore all timetables with more than 46 trips, knowing that it cannot appear as a block in an optimal solution. Of course, this is true for just the NL6 benchmark set; there is no guarantee that a similar result would hold for other distance matrices D. However, we make the conjecture that this 46-trip threshold actually does guarantee optimality for the large majority of distance matrices D.

Whenever this conjecture holds, part (c) of the algorithm would reduce from five hours to one minute. Therefore, it would take only two minutes (approximately 59 + 53 seconds) to create the matrices M and N representing the edge weights of the digraph G, while retaining the optimality of the final output.

Even if this conjecture were not true, there are several innovative techniques we can use to reduce the algorithm's running time in part (c). One such technique is to store expensive heuristic estimates in memory, so that the same calculations do not need to be repeated multiple times. In [19], the authors use this memory-storage technique to solve the double-round NL8 problem, reducing the running time of their depth-first-search algorithm from 26 hours to just 7 minutes on a single processor. Perhaps the same technique can be applied to reduce the running time of part (c) of our algorithm from five hours to several minutes.

Appendix B. Distance matrices

In this appendix, we explain how the distance matrices were calculated.

Table 7 listed the distance matrix for the Pacific League, for the Chiba Marines (M), Tohoku Eagles (E), Hokkaido Fighters (F), Orix Buffaloes (B), Fukuoka Hawks (H), and Saitama Lions (L). We now explain how each entry was derived. All distances are in kilometres.

(ME) Bus from Marine Stadium to Tokyo Station (34), shinkansen to Sendai Station (325), bus to Kleenex Stadium (2), for a total of 361.

(MF) Bus from Marine Stadium to Haneda Airport (42), airplane to Chitose Airport (822), bus to Sapporo Dome (40), for a total of 904.

Table 19
Distance matrix for the Central League (all distances in kilometres).

Team	Hiroshima	Hanshin	Chunichi	Yokohama	Yomiuri	Tokyo
Hiroshima	0	323	488	808	827	829
Hanshin	323	0	195	515	534	536
Chunichi	488	195	0	334	353	355
Yokohama	808	515	334	0	37	35
Yomiuri	827	534	353	37	0	7
Tokyo	829	536	355	35	7	0

(MB) Bus from Marine Stadium to Tokyo Station (34), shinkansen to Shin-Osaka Station (515), bus to Orix Stadium (33), for a total of 582.
(MH) Bus from Marine Stadium to Haneda Airport (42), airplane to Fukuoka Airport (881), bus to Yahoo Dome (11), for a total of 934.
(ML) Bus from Marine Stadium to Seibu Dome (86), for a total of 86.
(EF) Bus from Kleenex Stadium to Sendai Airport (20), airplane to Chitose Airport (520), bus to Sapporo Dome (40), for a total of 580.
(EB) Bus from Kleenex Stadium to Sendai Airport (20), airplane to Itami Airport (615), bus to Orix Stadium (35), for a total of 670.
(EH) Bus from Kleenex Stadium to Sendai Airport (20), airplane to Fukuoka Airport (1069), bus to Yahoo Dome (11), for a total of 1100.
(EL) Bus from Kleenex Stadium to Sendai Station (2), shinkansen to Tokyo Station (325), bus to Seibu Dome (47), for a total of 374.
(FB) Bus from Sapporo Dome to Chitose Airport (40), airplane to Itami Airport (1040), bus to Orix Stadium (35), for a total of 1115.
(FH) Bus from Sapporo Dome to Chitose Airport (40), airplane to Fukuoka Airport (1415), bus to Yahoo Dome (11), for a total of 1466.
(FL) Bus from Sapporo Dome to Chitose Airport (40), airplane to Haneda Airport (822), bus to Seibu Dome (66), for a total of 928.
(BH) Bus from Orix Stadium to Shin-Osaka Station (33), shinkansen to Hakata Station (554), bus to Yahoo Dome (8), for a total of 595.
(BL) Bus from Orix Stadium to Shin-Osaka Station (33), shinkansen to Tokyo Station (515), bus to Seibu Dome (47), for a total of 595.
(HL) Bus from Yahoo Dome to Fukuoka Airport (11), airplane to Haneda Airport (881), bus to Seibu Dome (66), for a total of 958.

Note: the Orix Buffaloes play half their games at Kyocera Dome, and half their games at Skymark Stadium. Itami Airport is located 20 kilometres from Kyocera and 50 kilometres from Skymark, for an average of 35 kilometres. Shin-Osaka Station is located 10 kilometres from Kyocera and 56 kilometres from Skymark, for an average of 33 kilometres. For simplicity, we denote Orix's home as "Orix Stadium", located 35 kilometres from the airport and 33 kilometres from the train station.

Table 19 provides the distance matrix for the Central League.

The six teams are the Hiroshima Carp (C), Hanshin Tigers (T), Chunichi Dragons (D), Yokohama Baystars (B), Yomiuri Giants (G), and the Tokyo Swallows (S). We now explain how each entry was derived. Once again, all distances are in kilometres. Note that the teams in the Central League travel to each intra-league venue via shinkansen instead of airplane as each of the cities are located on the same primary shinkansen line.

(CT) Bus from Municipal Stadium to Hiroshima Station (2), shinkansen to Shin Osaka Station (306), bus to Koshien Stadium (15), for a total of 323.
(CD) Bus from Municipal Stadium to Hiroshima Station (2), shinkansen to Nagoya Station (479), bus to Nagoya Dome (7), for a total of 488.
(CB) Bus from Municipal Stadium to Hiroshima Station (2), shinkansen to Shin Yokohama Station (796), bus to Yokohama Stadium (10), for a total of 808.
(CG) Bus from Municipal Stadium to Hiroshima Station (2), shinkansen to Tokyo Station (821), bus to Tokyo Dome (4), for a total of 827.
(CS) Bus from Municipal Stadium to Hiroshima Station (2), shinkansen to Tokyo Station (821), bus to Meiji Jingu Stadium (6), for a total of 829.
(TD) Bus from Koshien Stadium to Shin-Osaka Station (15), shinkansen to Nagoya Station (173), bus to Nagoya Dome (7), for a total of 195.
(TB) Bus from Koshien Stadium to Shin-Osaka Station (15), shinkansen to Shin Yokohama Station (490), bus to Yokohama Stadium (10), for a total of 515.
(TG) Bus from Koshien Stadium to Shin-Osaka Station (15), shinkansen to Tokyo Station (515), bus to Tokyo Dome (4), for a total of 534.
(TS) Bus from Koshien Stadium to Shin-Osaka Station (15), shinkansen to Tokyo Station (515), bus to Meiji Jingu Stadium (6), for a total of 536.
(DB) Bus from Nagoya Dome to Nagoya Station (7), shinkansen to Shin Yokohama Station (317), bus to Yokohama Stadium (10), for a total of 334.
(DG) Bus from Nagoya Dome to Nagoya Station (7), shinkansen to Tokyo Station (342), bus to Tokyo Dome (4), for a total of 353.
(DS) Bus from Nagoya Dome to Nagoya Station (7), shinkansen to Tokyo Station (342), bus to Meiji Jingu Stadium (6), for a total of 355.
(BG) Bus from Yokohama Stadium to Tokyo Dome (37), for a total of 37.
(BS) Bus from Yokohama Stadium to Meiji-Jingu Stadium (35), for a total of 35.
(GS) Bus from Tokyo Dome to Meiji-Jingu Stadium (7), for a total of 7.

Appendix C. Optimal schedules for the Central league

In Section 5, we presented the optimal schedule for the Pacific League, using the algorithm described in Section 4. In this appendix, we now repeat the process for the Central League, using the exact same algorithm. First, we present the optimal schedule in Table 20.

Under the 2010 NPB schedule, every team made at least 33 trips (see Table 21). Under our optimal schedule, no team would make any more than 29 trips (see Table 22).

Finally, we compare these two schedules with respect to the total distance traveled by all six teams, using our distance matrix.

Table 20
Optimal schedule for the Central League, total distance of 57,836 kilometers.

Team	R1	R2	R3	R4	R5	R6	R7	R8
Hiroshima Carp (C)	**TG**BSD	**BSD**TG	D**TGS**B	GS**BD**T	BT**DSG**	DSG**BT**	BT**DSG**	DSG**BT**
Hanshin Tigers (T)	C**DG**BS	G**BSC**D	BC**DGS**	DGS**BC**	D**CSG**B	SG**BD**C	D**CSG**B	SG**BD**C
Chunichi Dragons (D)	**B**TSG**C**	**SG**CB**T**	**C**STB**G**	**TB**GC**S**	**T**GC**BS**	**C**BST**G**	**T**GC**BS**	**C**BST**G**
Yokohama Baystars (B)	D**SCT**G	CT**GD**S	**T**GS**DC**	**S**DCT**G**	**C**SGD**T**	**GD**TC**S**	**C**SGD**T**	**GD**TC**S**
Yomiuri Giants (G)	**S**CT**DB**	**T**DBS**C**	**SB**CTD	**CTD**SB	S**DB**TC	B**TCS**D	S**DB**TC	B**TCS**D
Tokyo Swallows (S)	GB**DCT**	DCT**GB**	G**DB**CT	B**CTG**D	**GB**TCD	**TCD**GB	**GB**TCD	**TCD**GB

Table 21
Trips taken between Central League venues under the existing schedule.

	CT	CD	CB	CG	CS	TD	TB	TG	TS	DB	DG	DS	BG	BS	GS	Total
Hiroshima Carp (C)	5	8	5	5	3	0	1	1	1	0	0	0	0	2	2	33
Hanshin Tigers (T)	4	2	1	0	1	5	5	5	7	0	1	0	2	0	0	33
Chunichi Dragons (D)	0	7	0	0	1	5	0	1	2	5	5	4	2	1	0	33
Yokohama Baystars (B)	0	0	5	0	3	1	6	0	1	6	1	0	7	4	0	34
Yomiuri Giants (G)	1	0	1	6	0	2	1	3	1	1	5	0	5	0	7	33
Tokyo Swallows (S)	2	2	0	0	4	0	0	0	6	1	1	4	1	6	6	33
Total trips	12	19	12	11	12	13	13	10	18	13	13	8	17	13	15	199

Table 22
Trips taken between Central League venues under our optimal schedule.

	CT	CD	CB	CG	CS	TD	TB	TG	TS	DB	DG	DS	BG	BS	GS	Total
Hiroshima Carp (C)	4	4	3	2	1	0	3	1	0	0	1	3	1	1	3	27
Hanshin Tigers (T)	6	1	1	0	0	5	3	3	1	1	1	0	0	3	4	29
Chunichi Dragons (D)	0	4	1	3	0	3	1	0	4	4	4	1	0	2	1	28
Yokohama Baystars (B)	4	1	2	1	0	0	4	0	0	5	2	0	2	5	3	29
Yomiuri Giants (G)	4	0	2	2	0	1	0	3	0	1	5	1	2	3	4	28
Tokyo Swallows (S)	4	3	0	0	1	0	1	0	3	0	0	5	3	4	5	29
Total trips	22	13	9	8	2	9	12	7	8	11	13	10	8	18	20	170

Table 23
Comparison of existing and optimal schedules for the Central League.

	Distance (existing)	Distance (optimal)	% Reduction in distance	Trips (existing)	Trips (optimal)	% Reduction in trips
Hiroshima Carp	17,850	11,741	34.2	33	27	18.2
Hanshin Tigers	14,304	8712	39.1	33	29	12.1
Chunichi Dragons	11,790	11,665	1.1	33	28	15.2
Yokohama Baystars	13,104	8929	31.9	34	29	14.7
Yomiuri Giants	11,469	9020	21.4	33	28	15.2
Tokyo Swallows	10,550	7769	26.4	33	29	12.1
Total	79,067	57,836	26.8	199	170	14.6

Overall, our optimal schedule achieves a 26.8% reduction in total travel compared to the existing schedule, in addition to an 14.6% reduction in total trips taken. Table 23 lists the improvement on a team by team basis.

From Table 23, we see that *DistOptimal* is 57,836 and *DistExisting* is 79,067. Running our longest-path algorithm, we can show that *DistWorst* is 106,832. Therefore, the efficiency of the 2010 NPB Central League schedule is $\frac{106832-79067}{106832-57836} \sim 57\%$.

Despite the impressive 26.8% reduction in total distance traveled, we note that one team (Chunichi Dragons) achieves a paltry 1.1% reduction under our schedule. This discrepancy can be corrected by proposing a non-optimal feasible schedule with an objective value just slightly higher than 57,836 kilometres where the team-by-team percentage improvement is more equitably distributed.

References

[1] A. Anagnostopoulos, L. Michel, P.V. Hentenryck, Y. Vergados, A simulated annealing approach to the traveling tournament problem, Journal of Scheduling 9 (2006) 177–193.

[2] I. Anderson, R. Bailey, Completeness properties of conjugates of Latin squares based on groups, and an application to bipartite tournaments, Bulletin of the Institute of Combinatorics and its Applications 21 (1997) 95–99.

[3] K.K.H. Cheung, Solving mirrored traveling tournament problem benchmark instances with eight teams, Discrete Optimization 5 (2008) 138–143.

[4] K.K.H. Cheung, A Benders approach for computing improved lower bounds for the mirrored traveling tournament problem, Discrete Optimization 6 (2009) 189–196.

[5] F.N. Costa, S. Urrutia, C.C. Ribeiro, An ILS heuristic for the traveling tournament problem with fixed venues, in: Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, Montreal, 2008.

[6] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[7] K. Easton, G. Nemhauser, M. Trick, The traveling tournament problem: description and benchmarks, in: Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming, 2001, pp. 580–584.
[8] K. Easton, G. Nemhauser, M. Trick, Solving the travelling tournament problem: A combined integer programming and constraint programming approach, in: Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling, 2002, pp. 319–330.
[9] F.S. Hillier, G.J. Lieberman, Introduction to Operations Research, eighth ed., McGraw-Hill, 2004.
[10] S. Irnich, A new branch-and-price algorithm for the traveling tournament problem, European Journal of Operational Research 204 (2010) 218–228.
[11] G. Kendall, S. Knust, C.C. Ribeiro, S. Urrutia, Scheduling in sports: An annotated bibliography, Computers and Operations Research 37 (2010) 1–19.
[12] R. Lewis, J. Thompson, On the application of graph colouring techniques in round-robin sports scheduling, Computers and Operations Research 38 (2011) 190–204.
[13] A. Lim, B. Rodrigues, X. Zhang, A simulated annealing and hill-climbing algorithm for the traveling tournament problem, European Journal of Operational Research 174 (2006) 1459–1478.
[14] P. Rasmussen, M. Trick, A Benders approach for the constrained minimum break problem, European Journal of Operational Research 177 (2007) 198–213.
[15] P. Rasmussen, M. Trick, Round robin scheduling - a survey, European Journal of Operational Research 188 (2008) 617–636.
[16] C.C. Ribeiro, S. Urrutia, Heuristics for the mirrored traveling tournament problem, in: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, 2004, pp. 323–342.
[17] C. Thielen, S. Westphal, Complexity of the traveling tournament problem, Theoretical Computer Science 412 (4–5) (2011) 345–351.
[18] M. Trick, Challenge Traveling Tournament Problems. <http://mat.gsia.cmu.edu/TOURN/> (accessed 25.7.11).
[19] D.C. Uthus, P.J. Riddle, H.W. Guesgen, DFS* and the traveling tournament problem, in: Proceedings of the 6th International Conference on the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR 2009, Pittsburgh, Pennsylvania, 2009, pp. 279–293.
[20] W.D. Wallis, One-Factorizations, Kluwer Academic Publishers, 1997.