Graph Theory and Sports Scheduling

Richard Hoshino and Ken-ichi Kawarabayashi

Introduction
The effects of global warming have been well documented, especially in recent years. As a result, the majority of countries have made a commitment to reducing their greenhouse gas emissions, including many whose national governments have made ambitious and unrealistic promises. Meeting these targets will require a coordinated effort from policymakers, businesses, and large industries, and numerous creative solutions will need to be implemented to achieve the desired goal. One potential solution is based on discrete mathematics, where combinatorial and graph-theoretic techniques are applied to scheduling optimization, leading to economic and environmental benefits.

There are many practical roles for mathematically optimal schedules that reduce total travel distance, including supply-chain logistics and airplane flight assignments. In this paper we describe how to optimize the regular-season schedule for Nippon Professional Baseball (NPB), Japan’s most popular professional sports league, with annual revenues topping one billion U.S. dollars.

Given the authors’ background as graph theorists, this research was motivated by the innocent question of whether NPB scheduling could be converted into a much simpler shortest-path problem. As we describe in this paper, the answer to this question is affirmative. Consequently, we have succeeded in generating the distance-optimal NPB regular-season schedule which retains all of the league’s constraints that ensure competitive balance while reducing the total travel distance by 24.3%, or nearly 70,000 kilometers, as compared to the 2010 season schedule.

To solve the NPB scheduling problem, we have generalized and extended the Traveling Tournament Problem (TTP), a well-known topic in sports scheduling [10]. Our research has produced five papers, [4], [5], [6], [7], [8], describing the theoretical aspects of the problem, providing various heuristics for generating distance-optimal intra-league and inter-league schedules, and applying the results to optimize the NPB league schedule.

Shortly after introducing the Traveling Tournament Problem [2], Easton et al. formed a consulting company to develop schedules for professional sports leagues. Their company, the Sports Scheduling Group, has received the contract to produce the regular-season schedule for Major League Baseball in six of the past seven years. Having now completed all of our research on NPB scheduling, our hope is to obtain the contract to produce future NPB regular-season schedules. We are excited by the possibility of sharing our expertise and passion with Nippon Professional Baseball, working in partnership with the league to produce schedules that save money and reduce greenhouse gas emissions, thus making an important contribution to Japan, both economically and environmentally.
Table 1. The distance-optimal NPB schedule versus the actual 2010 regular-season schedule.

The Multi-Round Balanced Traveling Tournament Problem

Consider the intra-league schedule given in Table 2, which consists of \( n = 6 \) teams each playing \( k = 4 \) blocks of ten sets. Each block consists of two rounds, with each round having \( n - 1 = 5 \) sets. Thus, each team plays a total of \( k(2n - 2) = 40 \) sets.

In this schedule,\(^1\) as with all subsequent schedules presented in this paper, home sets are marked in red.

Let \( n \) and \( k \) be positive integers. 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 \( 1 \leq i, j \leq n \), and all diagonal entries \( D_{i,i} \) are zero.

For any pair \((n,k)\) and distance matrix \( D \), the solution to the Multi-Round Balanced Traveling Tournament Problem (mb-TTP) is an intra-league tournament schedule that minimizes the total distance traveled by all \( n \) teams, subject to the following conditions:

(a) The compactness condition: The tournament lasts \( k(2n - 2) \) sets, i.e., \( 2k \) rounds, where each team has one set scheduled in each time slot. (Thus \( n \) must be even.)

(b) The each-round condition: Each pair of teams must play exactly once per round, with their matches in rounds \( 2t - 1 \) and

---

\(^1\)For example, team \( p_1 \) starts by playing a home set against \( p_2 \), followed by three consecutive road sets against \( p_5, p_3, p_4 \), then returning home to play three consecutive home sets against \( p_6, p_3, p_4 \), and so on.
2t taking place at different venues (for all $1 \leq t \leq k$).
(c) The at-most-three condition: No team may have a home stand or road trip lasting more than three sets.
(d) The no-repeat condition: A team cannot play against the same opponent in two consecutive sets.
(e) The diff-two condition: Let $H_{i,s}$ and $R_{i,s}$ be the number of home and away sets played by team $i$ within the first $s$ sets. Then $|H_{i,s} - R_{i,s}| \leq 2$ for all $(i,s)$ with $1 \leq i \leq n$ and $1 \leq s \leq k(2n - 2)$.

For example, one can quickly verify that Table 2 satisfies all five conditions. When calculating the total distance, we will assume that each team begins the tournament at home and returns home after having played their last away set. Furthermore, when a team is scheduled for a road trip consisting of multiple away sets, the team doesn’t return to their home city but rather proceeds directly to their next away venue.\(^2\) The mb-TTP is an extension of the well-known NP-hard Traveling Salesman Problem, asking for an optimal schedule linking venues that are close to one another.

We now present an algorithm for solving the mb-TTP, for any $k \geq 1$, 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 \([1]\) 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 nonnegative edge weights.

**Shortest-Path Reformulation**

By definition, a block is a two-round tournament schedule satisfying the conditions of the mb-TTP,

<table>
<thead>
<tr>
<th>Team</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>$p_2p_3p_4p_5$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_2p_4p_5p_3$</td>
<td>$p_6p_5p_4p_3$</td>
<td>$p_3p_4p_5p_2$</td>
<td>$p_6p_5p_4p_3$</td>
<td>$p_3p_4p_5p_2$</td>
<td></td>
</tr>
<tr>
<td>$p_2$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_4p_5p_3p_2$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_4p_5p_2$</td>
<td>$p_6p_5p_4p_3$</td>
<td>$p_3p_4p_5p_2$</td>
<td>$p_1p_2p_6p_4$</td>
<td></td>
</tr>
<tr>
<td>$p_3$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td></td>
</tr>
<tr>
<td>$p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td></td>
</tr>
<tr>
<td>$p_5$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td></td>
</tr>
<tr>
<td>$p_6$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td>$p_1p_2p_6p_4$</td>
<td>$p_3p_2p_6p_4$</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. A 40-set (120-game) intra-league schedule for the NPB Pacific League.

with each of the $n$ teams playing $2(n - 1)$ sets of games. 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 multiblock schedule without violating any of the five conditions of the mb-TTP. As we will explain, to determine whether two (feasible) 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 $B_2$.

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, there are $\binom{n}{n/2}$ ways to select the home teams. Also, there are $\binom{n}{n/2} \cdot \left(\frac{n}{2}\right)!$ ways to specify the matches of any column, since there are $\left(\frac{n}{2}\right)!$ ways to map any choice 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 $m = \binom{n}{n/2}^2 \cdot \binom{n}{n/2}$ 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-to-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.
Figure 2. Reformulation of the $k$-block mb-TTP as a shortest-path problem.

For example, $(p_5, p_3, p_2, p_6, p_1, p_4)^T$ is one of the 120 possibilities for the matches column, and $(0, 1, 1, 0, 1, 0)^T$ 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 $\frac{120}{2} = 60$ unique columns, corresponding to the 15 perfect matchings of the complete graph $K_6$.

For the NPB Pacific League, there exists some integer $q$ (with $1 \leq q \leq 2400$) that is the index of the instance where the home teams column is $(0, 1, 1, 0, 1, 0)^T$ and the matches column is $(p_5, p_3, p_2, p_6, p_1, p_4)^T$. Similarly, there exists some $r$ (with $1 \leq r \leq 2400$) that is the index of the instance where the two columns are $(p_2, p_1, p_6, p_5, p_4, p_3)^T$ and $(1, 1, 0, 0, 0, 1)^T$. In Table 2 the last two columns of block 1 have index $q$ and the first two columns of block 2 have index $r$.

For each pair $(u_1, u_2)$, with $1 \leq u_1, u_2 \leq m$, 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 $x$ and $y$ from the previous paragraph, we have $C_{q,r}$.

\[
\begin{bmatrix}
0 & p_5 & p_2 & 1 \\
p_3 & p_1 & 1 \\
p_2 & p_6 & 0 \\
p_6 & p_5 & 0 \\
p_1 & p_4 & 0 \\
p_4 & p_3 & 1
\end{bmatrix}
\]

Note that $C_{q,r}$ 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. As we describe in Theorem 1, these three properties are a necessary and sufficient condition for whether two feasible blocks can be concatenated to produce a multiblock schedule satisfying the conditions of the mb-TTP.

Before we proceed with Theorem 1, let us 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 $v_{\text{start}}$, a sink vertex $v_{\text{end}}$, and vertices $x_{t,u}$ and $y_{t,u}$ defined for each $1 \leq t \leq k$ and $1 \leq u \leq m$.

We now describe how these edges are connected with a pictorial representation of $G$ in Figure 2. For notational simplicity, denote $v_1 \rightarrow v_2$ as the directed edge from $v_1$ to $v_2$.

(i) For each $1 \leq u \leq m$, add the edge $v_{\text{start}} \rightarrow x_{1,u}$.

(ii) For each $1 \leq u \leq m$, add the edge $y_{k,u} \rightarrow v_{\text{end}}$.

(iii) For each $1 \leq t \leq k$ and for each $1 \leq u_1, u_2 \leq m$, add the edge $x_{t,u_1} \rightarrow y_{t,u_2}$ if there exists a (feasible) block for which the first two columns have index $u_1$ and the last two columns have index $u_2$.

(iv) For each $1 \leq t \leq k - 1$ and for each $1 \leq u_1, u_2 \leq m$, add the edge $y_{t,u_2} \rightarrow x_{t+1,u_1}$ if 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 [4] shows that the $k$-block mb-TTP can be reformulated in a graph-theoretic context for any $k \geq 1$.

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

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 $v_{\text{start}}$ to $v_{\text{end}}$ 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:
Table 3. The distance-optimal intra-league schedule for the NPB Central League.

<table>
<thead>
<tr>
<th>Team</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
</tr>
</thead>
<tbody>
<tr>
<td>c₁</td>
<td>c₂c₅c₄c₆c₃</td>
<td>c₄c₂c₃c₅c₄</td>
<td>c₅c₆c₁c₂c₃</td>
<td>c₆c₁c₂c₃c₄</td>
<td>c₁c₆c₅c₃c₂</td>
<td>c₅c₁c₄c₆c₂</td>
<td>c₆c₂c₁c₃c₄</td>
<td>c₂c₁c₃c₅c₄</td>
</tr>
<tr>
<td>c₂</td>
<td>c₁c₃c₅c₄c₆</td>
<td>c₅c₁c₂c₃c₄</td>
<td>c₆c₅c₁c₂c₃</td>
<td>c₁c₆c₅c₄c₂</td>
<td>c₆c₃c₄c₂c₁</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₁c₊c₆c₅c₄</td>
<td>c₃c₄c₂c₁c₆</td>
</tr>
<tr>
<td>c₃</td>
<td>c₄c₆c₅c₁c₂</td>
<td>c₅c₆c₁c₂c₃</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₄c₃c₁c₂c₆</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₆c₅c₁c₂</td>
</tr>
<tr>
<td>c₄</td>
<td>c₆c₅c₁c₂c₃</td>
<td>c₅c₆c₁c₂c₃</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₄c₃c₁c₂c₆</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₆c₅c₁c₂</td>
</tr>
<tr>
<td>c₅</td>
<td>c₆c₁c₂c₃c₄</td>
<td>c₅c₆c₁c₂c₃</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₄c₃c₁c₂c₆</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₆c₅c₁c₂</td>
</tr>
<tr>
<td>c₆</td>
<td>c₁c₆c₅c₃c₂</td>
<td>c₅c₆c₁c₂c₃</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₃c₄c₁c₂c₆</td>
<td>c₄c₃c₁c₂c₆</td>
<td>c₆c₃c₂c₁c₄</td>
<td>c₃c₆c₅c₁c₂</td>
</tr>
</tbody>
</table>

(a) For each 1 ≤ u ≤ m, the weight of edge v_start → x₁,u is the total distance traveled by the 2 teams making the trip from their home city to the venue of their opponent in set 1.

(b) For each 1 ≤ u ≤ m, the weight of edge yₖ,u → v_end is the total distance traveled by the 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 ≤ t ≤ k, and for each 1 ≤ u₁, u₂ ≤ m, the weight of edge xₜ,u₁ → yₚ,u₂ is the minimum in-distance of a block, selected among all blocks for which the first two columns have index u₁ and the last two columns have index u₂.

(d) For each 1 ≤ t ≤ k − 1 and for each 1 ≤ u₁, u₂ ≤ m, the weight of edge yₜ,u₂ → xₜ₊₁,u₁ is the total distance traveled by the teams that travel from their match in set 2t(n − 1) to their match in set 2(t + 1) + 1, where the last two columns of the tth block have index u₂ and the first two columns of the (t + 1)th block have index u₁.

To illustrate (d), consider the first two blocks in Table 2, where the last two columns of block 1 have index q and the first two columns of block 2 have index r. When we concatenate these two blocks, the weight of edge y₁,q → x₂,r is the total distance traveled by the teams from their matches in set 10 to their matches in set 11. This sum equals D_p₁,q₁ + D_p₂,q₂ + D_p₃,q₃, the distances traveled by teams p₂, p₃, and p₄, 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 u₁ and the last two columns have index u₂. 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 x₁,u₁ → yₚ,u₂, for all 1 ≤ u₁, u₂ ≤ m. As a result, we have a digraph G on 2mk + 2 vertices and at most 2m + (2k − 1)m² edges, with a unique weight for each edge. Combined with the previous theorem, we have established the following.

Theorem 2. Let P = v_start → x₁,p₁ → y₁,q₁ → x₂,p₂ → y₂,q₂ → · · · → xₖ,pₖ → yₖ,qₖ → 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 ≤ t ≤ k, let Bₜ be the block of minimum in-distance selected among all blocks for which the first two columns have index pₜ and the last two columns have index qₜ. Then the multiblock schedule S = B₁, B₂, . . . , Bₖ, created by concatenating the k blocks consecutively, is an optimal solution of the mb-TTP.

Application to the NPB

Therefore, we have shown that the mb-TTP is isomorphic to finding the shortest weighted path in the directed graph G. For the case n = 6, we can show that G consists of 4800k + 2 vertices and 2400 + 2400 + 2618520 + 1486320(k − 1) = 4104840 − 1481520 edges. Given the distance matrices for the NPB Pacific and Central Leagues, we can determine the appropriate weights for each edge in G to determine the shortest path from v_start to v_end using Dijkstra’s Algorithm, which in turn generates the distance-optimal schedule. We wrote our code using Maplesoft; for each league, Maplesoft produced the distance-optimal intra-league schedule after five hours of computation.
time. The large majority of the running time was spent determining the correct edge weight for part (c) of our construction. For more information, we refer the reader to [7].

The optimal Pacific League intra-league schedule appears in Table 2; the optimal Central League intra-league schedule appears in Table 3.

In Table 1 we produced an overall summary of the results. In Table 4 we provide a team-by-team breakdown for the NPB Pacific League, comparing our distance-optimal schedule to the actual schedule played by the teams during the 2010 regular season.

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 intra-league schedule, team $p_1$ traveled nearly 12,500 kilometers more than team $p_3$. Under our schedule, the difference between the most traveled and least traveled would reduce to just 4,500 kilometers. For the Central League, the difference between the most traveled and least traveled would reduce from 7,500 kilometers to just 4,000 kilometers.

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” that need to be scheduled on particular days to optimize revenue and that are often dictated by the wishes of television broadcasters 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 the best possible for all parties involved.

Epilogue

After the publication of our initial results, the Nippon Professional Baseball (NPB) League invited us (the authors) to visit their head office. Over the course of three meetings in September 2012, we consulted the NPB and helped them design the Central League’s 2013 intra-league schedule [3].

When we met with the NPB head scheduler, we learned that the league has additional scheduling constraints, such as the “revenue-balancing” requirement that each team play the same number of weekend home games, weekday home games, weekend road games, and weekday road games. In [9], we describe how we fully solved this TTP-variant for the case $n = 6$, and helped the NPB design the intra-league schedule for the 2013 Central League, reducing the number of total trips by 12 and cutting over six thousand kilometers of total travel distance, as compared to the schedule from the previous season.

We look forward to partnering with the NPB once again, and hope to have the opportunity to help this league produce future regular-season schedules that will result in annual win-wins for the people of Japan, both economically and environmentally.

References