

Complexity of the Traveling Tournament Problem

Clemens Thielen and Stephan Westphal

Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14,
D-67663 Kaiserslautern, Germany
{thielen,westphal}@mathematik.uni-kl.de

Abstract. We consider the complexity of the traveling tournament problem, which is a well-known benchmark problem in tournament timetabling. The problem was supposed to be computationally hard ever since its proposal in 2001. Recently, the first NP-completeness proof has been given for the variant of the problem where no constraints on the number of consecutive home games or away games of a team are considered. The complexity of the original traveling tournament problem including these constraints, however, is still open. In this paper, we show that this variant of the problem is strongly NP-complete when the upper bound on the maximal number of consecutive away games is set to 3.

Key words: traveling tournament problem, timetabling, computational complexity

1 Introduction

Professional sports leagues exist all over the world. Popular leagues are often of huge economic importance due to the enormous revenues generated by selling tickets and broadcasting rights for the games. Hence, the planning of these leagues is of major importance. An important aspect is the generation of a timetable for the tournaments that specifies the order in which the teams play each other during the season and the venue of each game. A well-studied variant of this problem is the traveling tournament problem (TTP), which was formally introduced by Easton et al.[1] in 2001. Given the number of teams and the pairwise distances between their home venues, TTP asks for a timetable of a double round robin tournament that minimizes the sum of the distances traveled by the teams during the season. This problem is quite important in practice, for example in the US, where the distances between two teams' home venues are often quite large, so minimizing travel distance for the teams becomes a major issue.

Ever since its proposal, TTP was supposed to be computationally hard to solve due to the similarity to the traveling salesman problem (TSP), which is known to be strongly NP-hard to solve and has become an important benchmark problem in computer science. Recently, the first NP-completeness proof

for a variant of TTP has been given by Bhattacharyya [2]. In this variant, no bounds on the number of consecutive home games or away games of a team are considered, which allows a reduction from TSP. The original traveling tournament problem including upper and lower bounds on the number of consecutive home games or away games, however, is different in nature since the main problem here is to find good tours for each team containing not more than the given maximum number of games and to synchronize these in order to obtain a feasible schedule. Hence, the reduction from TSP used in [2] is not applicable and the complexity of the original TTP still remains open.

The complexity of the original TTP with consecutive home/away bounds is an important question as this is the most relevant variant of TTP in practice. In real world sports leagues, schedules in which some teams have only away games (or only home games) for very long time periods are not considered feasible as such schedules do not generate continuous revenue due to ticket sales for the teams. Moreover, schedules in which the home and away games are distributed in a more uniform way lead to more equitable team-rankings during the season. For these reasons, the schedules of most major sports leagues, e.g., the Major League Baseball (MLB) in the US, contain restrictions on the number of consecutive home and away games. The most common restriction used is an upper bound of 3 consecutive home games or away games, which is exactly the case we consider in this paper.

We now formally define the traveling tournament problem (TTP) and introduce our notation. We are given a set T of teams, where $|T| = n \geq 4$ is even. An $(n \times n)$ -distance matrix $D = (d_{ij})$ specifies the distances between the home venues of the teams, i.e., $d_{ij} \geq 0$ is the distance between the home venues of teams i and j . The distances are assumed to be symmetric (i.e., $d_{ij} = d_{ji}$ for all i, j) and satisfy $d_{ii} = 0$ for all i as well as the triangle inequality (i.e., $d_{ij} + d_{jk} \geq d_{ik}$ for all i, j, k). A *game* is an ordered pair of teams, where the first team is the home team and the second the away team. A sequence of consecutive away games of a team is called a *road trip*, and a sequence of consecutive home games is called a *home stand*. A *double round robin tournament* is a collection of games in which every team plays every other team once at home and once away (i.e., at the other team's home venue). Hence, exactly $2n - 2$ time slots are necessary for a double round robin tournament. Before the tournament, each team is assumed to stay at its home venue and it has to return there after the tournament in case that its last game is an away game. Between two consecutive away games, a team travels directly from the venue of the first opponent to the venue of the second opponent.

With this terminology, the traveling tournament problem for a positive integer $k \geq 3$ is defined as follows:

Definition 1 (The Traveling Tournament Problem (TTP(k))). ([1])

INSTANCE: The set T of teams, the distance matrix $D = (d_{ij})$, and a nonnegative integer $l \geq 0$.

QUESTION: Does there exist a double round robin tournament of the teams in T satisfying the following conditions:

- (a) The length of any home stand is at most k .
- (b) The length of any road trip is at most k .
- (c) Game j at i is not followed immediately by game i at j .
- (d) The sum of the distances traveled by the teams is at most l .

Note that we stated the problem in its decision version as we study its computational complexity.

2 Previous Work

Since the proposal of TTP by Easton et al. [1], most work on the problem focused on the design of approximation algorithms and heuristics for the problem (cf., for example, [3–5]). The first algorithm with a constant approximation ratio was the $(2 + (9/4)/(n - 1))$ -approximation algorithm for TTP(3) proposed by Miyashiro et al. [6]. Recently, Yamaguchi et al. [7] presented an approximation algorithm for TTP(k) whose approximation ratio is bounded by $(2k - 1)/k + \mathcal{O}(k/n)$ when $k \leq 5$ and by $(5k - 7)/(2k) + \mathcal{O}(k/n)$ when $k > 5$. Surveys on round robin scheduling and TTP can be found in [8, 9].

3 Our Contribution

We show that the decision problem TTP(3) is strongly NP-complete. To show NP-hardness, we present a reduction from 3-satisfiability (3-SAT). This reduction shows that TTP(3) is strongly NP-hard even when restrictions (a) on maximum length of home stands and (c) on consecutive games between the same teams are removed, as our reduction does not rely on these two restrictions. We also suspect that the techniques used in our proof can be generalized to show that TTP(k) is NP-complete for fixed $k > 3$.

4 Proof of NP-Completeness

In this section, we proof NP-completeness of TTP(3). To show NP-hardness, we use a reduction from 3-satisfiability (3-SAT), the well-known NP-complete problem of deciding whether a boolean formula in conjunctive normal form with three literals per clause admits a satisfying assignment (cf., for example, [10]). In the following, the set of variables of a 3-SAT instance will be denoted by $\{x_1, \dots, x_n\}$ and the set of clauses by $\{C_1, \dots, C_p\}$.

The following easy result shows that 3-SAT remains NP-complete even if restricted to instances in which the number of clauses is a multiple of 6 and each literal occurs exactly as often as its negation (such instances will be used in our reduction):

Lemma 1. *3-SAT remains strongly NP-hard if the number of clauses in an instance is restricted to multiples of 6 and the number of occurrences of x_i is equal to the number of occurrences of \bar{x}_i for every $i = 1, \dots, n$.*

Proof. Denote the number of occurrences of x_i, \bar{x}_i by n_i and \bar{n}_i , respectively. If $n_i - \bar{n}_i =: l > 0$ for some i , we add clauses $C_{p+1} = C_{p+2} = \dots = C_{p+l} = (\bar{x}_i \vee x_{n+1} \vee \bar{x}_{n+1})$ for a new variable x_{n+1} . This does not affect satisfiability of the given formula since C_{p+1}, \dots, C_{p+l} are satisfied in any truth assignment, so we obtain an equivalent instance in which $n_i = \bar{n}_i$. To make the number p of clauses a multiple of 6, observe that the number of clauses in a formula with $n_i = \bar{n}_i$ for all i must always be even. Hence, by adding at most 2 pairs of clauses of the form $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_1 \vee \bar{x}_2)$ we can increase the number of clauses to the next multiple of 6 while not affecting satisfiability and the property that $n_i = \bar{n}_i$ for all i . \square

We now show our main result, which states that TTP(3) is strongly NP-complete. Our proof contains a graph construction similar to the one used by Itai et al. [11], who showed that deciding whether an undirected graph contains a given number of vertex-disjoint s - t -paths of a specified length is NP-complete.

Theorem 1. *TTP(3) is strongly NP-complete.*

Proof. Membership in NP is obvious as the distance traveled by the teams in a given tournament can easily be calculated in polynomial time. To show NP-hardness, we show that 3-SAT is polynomial time reducible to TTP(3). Let $\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_p$ be an instance of 3-SAT with $p/6 \in \mathbb{N}$ and $n_i = \bar{n}_i$ for $i = 1, \dots, n$. We construct a complete, undirected graph on a set V of vertices with edge costs d_{ij} , $i, j \in V$, such that φ is satisfiable if and only if the instance of TTP(3) with teams situated at the vertices of the graph (possibly more than one team per vertex) and distance matrix $D = (d_{ij})$ given by the edge costs has a solution with cost (overall distance traveled) at most a certain value. This complete, undirected graph is constructed from an undirected graph $G = (V, E)$ by adding all edges that are not present in G . The idea behind this construction is that the edges that are not in E have higher costs, so that, in a cheap tournament, the teams mostly travel on the edges in E .

We now describe the construction of the graph G . Afterwards, we will define the costs of the edges. G contains n subgraphs G_1, \dots, G_n which only have one vertex t in common. The subgraph G_i shown in Figure 1 is associated with variable x_i in the formula φ . For each k , the vertices $x_{i,k}, \bar{x}_{i,k}$ correspond to the k -th occurrence of x_i and \bar{x}_i in φ , respectively. The total number of vertices in G_i (not counting t) is $4n_i$ ($2n_i$ vertices $x_{i,j}$, n_i vertices $w_{i,j}$, and n_i vertices $y_{i,j}$).

In addition to G_1, \dots, G_n , the graph G contains two vertices c_l and z_l for every clause C_l in φ and the edge $[c_l, z_l]$ for every $l = 1, \dots, p$. Moreover, if the

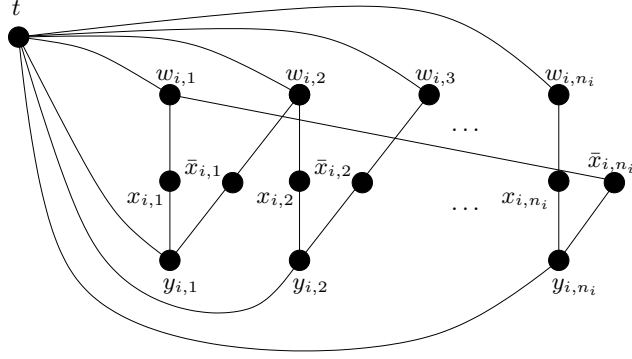


Fig. 1. Subgraph G_i associated with variable x_i

k -th occurrence of x_i (\bar{x}_i) is in clause C_l , there is an edge between x_{ik} (\bar{x}_{ik}) and c_l . Vertex c_l corresponds to the clause C_l and vertex z_l serves as an intermediate vertex to make sure that cycles through c_l have the correct length. Figure 2 shows the graph G associated with

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_4) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_4),$$

where we chose a small example with p not a multiple of 6 for illustration purposes.

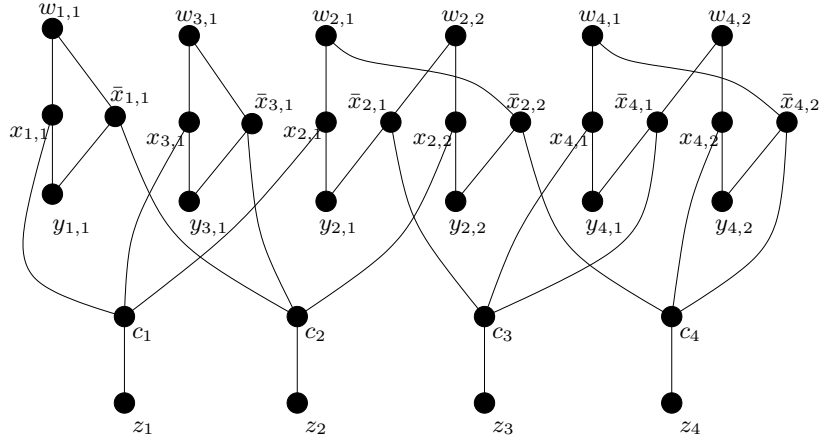


Fig. 2. Graph G for the example without the edges from t to all $x_{i,j}, w_{i,j}, y_{i,j}, z_i$

The number of additional vertices in G (in addition to the vertices contained in the G_i) is $2p$ (c_l and z_l for each $l = 1, \dots, p$), so the total number of vertices

in G when not counting t is

$$2p + \underbrace{\sum_{i=1}^n (4n_i)}_{=6p} = 8p =: m.$$

The costs the edges are set as follows: All edges that are contained in the graph G first get a cost of 1 and all other edges in the complete graph on V get cost 2. Then the costs of all edges in the complete graph that are incident to t are increased by $(M - 1)$, where M is a large integer whose exact value will be specified later. Finally, the cost of all edges $[t, c_l]$ is further increased by 1. Hence, in the final graph, the vertices $w_{i,j}$ and $y_{i,j}$ are connected to t by edges of cost M , the vertices $x_{i,j}$, $\bar{x}_{i,j}$, and z_l by edges of cost $(M + 1)$, and the vertices c_l by edges of cost $(M + 2)$.

The teams in our TTP-instance are situated as follows: One team is placed at each vertex in V except the vertex t , at which m^3 teams are placed (i.e., all these teams have distance 0 to each other and the same distance to the teams outside t). In the following, we will refer to the vertices in $S := V \setminus \{t\}$ as the *small* vertices. Observe that the distances between the teams given by the edge costs of the graph satisfy the triangle inequality: First, the costs before adding $(M - 1)$ to the cost of all edges incident to t are all in $\{1, 2\}$, and such costs always satisfy the triangle inequality. Second, adding the constant $(M - 1)$ to all edges incident to the vertex t cannot invalidate the triangle inequality. Last, further increasing the cost of the edges $[t, c_l]$ by 1 does also not invalidate the triangle inequality as the vertices c_l are only connected via edges of cost 1 to vertices $x_{i,j}$, $\bar{x}_{i,j}$, and z_l , which are connected to t by edges of cost $(M + 1)$.

We now derive bounds on the distances traveled by all teams at the small vertices in any tournament in order to show that the costs of a tournament are dominated by the distances traveled by the teams in t .

The distance that any team at a small vertex travels for visiting all the other teams at small vertices is at most $4(m - 1)$ since it can visit every one of the $(m - 1)$ other teams separately (i.e. do only tours including a single match) and such a tour has cost at most 4 since each edge between the small vertices has cost at most 2. Hence, the overall distance traveled by the teams at the small vertices for visiting each other is at most

$$4 \cdot m \cdot (m - 1) \leq 4m^2. \tag{1}$$

The distance that a team at a small vertex travels in any tournament for visiting the teams in t is at least $2 \cdot (m^3/3)$ times the cost of the edge connecting its vertex to t : At least $(m^3/3)$ tours of length at most 3 are necessary to visit all the m^3 teams in t and any such tour has to enter and leave t , each of which cannot be done at a cost lower than the cost of the edge connecting its vertex to t . Since the p vertices c_l are connected to t by edges of cost $(M + 2)$, the

p vertices z_l and the $3p$ vertices $x_{i,j}/\bar{x}_{i,j}$ by edges of cost $(M + 1)$, and the $3p$ vertices $w_{i,j}, y_{i,j}$ by edges of cost M , this implies that the overall distance traveled by the teams at the small vertices for visiting the teams in t in any tournament is at least

$$2(m^3/3) \cdot \left(p \cdot (M + 2) + 4p \cdot (M + 1) + 3p \cdot M \right) = 2(m^3/3) \cdot (8pM + 6p). \quad (2)$$

When considering the distance traveled by the teams in t for visiting the teams at the small vertices, observe that any tour from t to some of the small vertices contains at least two edges of cost $\geq M$ as the tour must leave t and enter t again, both of which is only possible by using edges of cost $\geq M$. When choosing M large enough compared to m (e.g., $M = m^5$), we can be sure that, in any optimal tournament, any team in t will only use the minimal possible number of tours to the small vertices. Since we will show that there is in fact a tournament in which every team in t uses only $m/3$ tours to vertices outside t , it follows that, in any optimal tournament, the teams in t only use tours of length 3 to visit teams at the small vertices.

We now show that our instance of TTP(3) admits a tournament of total cost at most ζ if and only if φ is satisfiable, where

$$\zeta := (16/3)M \cdot p \cdot (m^3 + 1) + 4m^2 \cdot (m \cdot p + 1) + 8p.$$

“ \Leftarrow ”: First assume that φ is satisfiable and let $t(x_i) \in \{\text{true}, \text{false}\}$ denote the truth value of variable x_i in a satisfying assignment. We construct a tournament of cost at most ζ . To do so, we first construct tours used by the teams in t for visiting the teams at the small vertices. At the same time, we show that these tours are as cheap as possible and every optimal tournament must use tours of a similar structure.

To this end, we first consider the minimal cost of a set of $m/3$ node-disjoint tours of a team in t in which all the vertices $x_{i,j}, \bar{x}_{i,j}, c_l$, and z_l are visited (possibly leaving the vertices $w_{i,j}$ and $y_{i,j}$ unvisited for the moment). First observe that, since there are $5p$ vertices to visit ($3p$ vertices $x_{i,j}, \bar{x}_{i,j}$, p vertices c_l , and p vertices z_l) by exactly $m/3 = (8/3)p$ tours, we need to use exactly $5p - (8/3)p = (7/3)p$ edges between small vertices in the tours. Hence, the cheapest way to cover all vertices $x_{i,j}, \bar{x}_{i,j}, c_l$, and z_l by exactly $m/3$ node-disjoint tours is to use as many edges of cost 1 (i.e., edges of G) as possible while at the same time using as few as possible of the edges $[t, c_l]$ of cost $(M + 2)$ as connections to t . The only way to cover as many vertices as possible in this fashion is to use a tour with 2 edges for every one of the p stars with centers c_l into which the graph G decomposes after deleting all nodes $w_{i,j}, z_{i,j}$ (see Figure 3).

As each such tour in one of the stars has cost $2(M+1)+2$, we may assume that each of the tours we choose contains an edge $[c_l, z_l]$. Moreover, we may choose the x -vertices contained in the star-tours as follows: We only choose vertex $x_{i,j}$ ($\bar{x}_{i,j}$) if the truth value $t(x_i)$ of variable x_i in the satisfying assignment for φ chosen above is false (true).

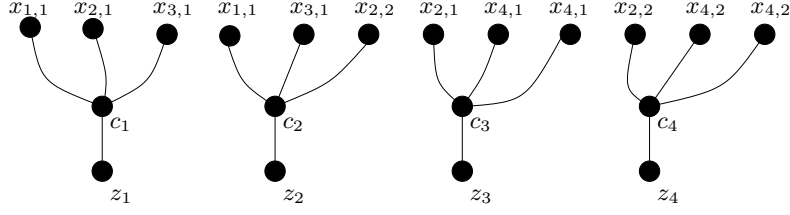


Fig. 3. The graph G decomposes into stars when all vertices $w_{i,j}, y_{i,j}$ are deleted.

After choosing the p tours in the stars, our total cost sums up to $p \cdot (2(M+1) + 2)$, all vertices c_l and z_l are covered, and p of the vertices $x_{i,j}/\bar{x}_{i,j}$ are covered. Hence, it remains to cover the $2p$ remaining x -vertices by exactly $(5/3)p$ tours. To do so, we must use exactly $2p - (5/3)p = (1/3)p$ additional edges between small vertices. As all these vertices are x -vertices, all the edges connecting them are of cost 2, so the cost is not affected by the choice of these $(1/3)p$ edges. In particular, the following choice has lowest possible cost: We choose $(3/2)p$ x -vertices that are not yet covered such that, in the original graph G , each $w_{i,j}$ is connected to exactly one of them (this is possible since every $w_{i,j}$ is connected to vertices $x_{i,j}$ and $\bar{x}_{i,k}$ at most one of which was already covered so far). These $(3/2)p$ x -vertices will each be covered by a separate tour that does not cover any other vertices. Each such tour has cost $2(M+1)$. The remaining $p/2$ x -vertices will be covered by the remaining $(1/6)p$ tours, each of which covers 3 x -vertices and has cost $2(M+1) + 4$. The final set of tours for the example (without the tours covering 3 x -vertices) is shown in Figure 4. Its total cost is

$$\begin{aligned}
 & \underbrace{p \cdot (2(M+1) + 2)}_{p \text{ tours } (t,x,c,z,t)} + \underbrace{(3/2)p \cdot (2(M+1))}_{(3/2)p \text{ tours } (t,x,t)} + \underbrace{(1/6)p \cdot (2(M+1) + 4)}_{(1/6)p \text{ tours } (t,x,x,x,t)} \\
 &= p \cdot ((16/3)M + 8),
 \end{aligned}$$

which, by our argumentation, is the lowest possible cost for $(m/3)$ tours covering all vertices $x_{i,j}, \bar{x}_{i,j}, c_l$, and z_l . In particular, every set of tours covering *all* the small vertices must have at least this cost.

We now show that we can extend our set of tours for the teams in t obtained so far such that it also covers all the vertices $w_{i,j}$ and $y_{i,j}$ while at the same time not changing its cost. In particular, this shows that the extension yields an optimal set of tours that can be used by the teams in t for visiting the small vertices.

The extension works as follows: The $(3/2)p$ x -nodes that are so far covered by tours of the form (t, x, t) of cost $2(M+1)$ per tour were chosen such that each $w_{i,j}$ (and, thus, also each $y_{i,j}$) is connected to one of them. Hence, by extending each tour of the form (t, x, t) to a tour of the form (t, w, x, y, t) , we can cover all vertices $w_{i,j}, y_{i,j}$ without introducing any additional tours. Moreover, each extended tour uses two edges of cost M ($[t, w]$ and $[y, t]$) and two edges of cost 1

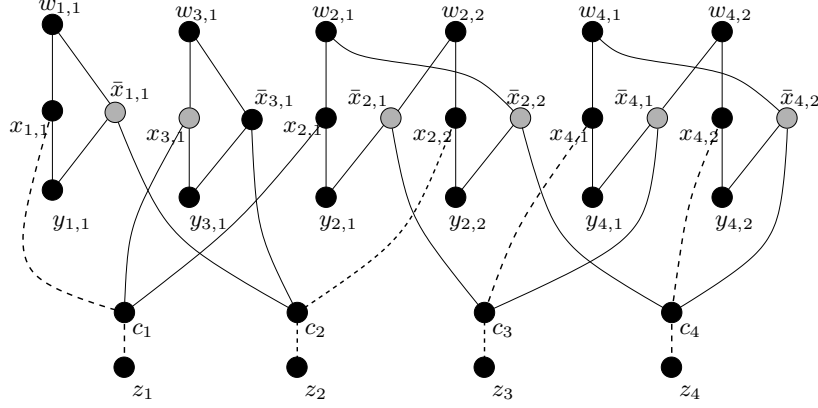


Fig. 4. The final set of tours for the example. x -vertices on separate tours are colored gray.

$([w, x]$ and $[x, y])$, so its total cost sums up to $2M + 2$, which equals the cost of the tour before the extension.

It remains to show how we can use the set of tours for the teams in t just constructed in a feasible tournament. To this end, let W be the set of teams located at t and let S be the set of teams at the small vertices.

Given the $m/3$ tours of length 3 for the teams in t constructed above, we now construct a tournament such that each team located at t visits the teams at the small vertices by taking exactly these trips. Thus, there is a set of trips $r_1, r_2, \dots, r_{m/3}$ with every r_j defined by a set of small vertices $s_{1,j}, s_{2,j}, s_{3,j}$ such that $s_{i,j}$ is the i th team visited in trip j . At the beginning of the tournament, there are $m/3$ big teams $t_{1,1}, t_{1,2}, \dots, t_{1,m/3}$ which travel these tours. For $j \in \{1, 2, \dots, m/6\}$, $t_{1,j}$ visits the team $s_{1,j}$, then plays against $s_{2,j}$, and finally concludes its tour by visiting $s_{3,j}$. Afterwards, it is visited by these three teams in the same order, and then starts off to visit the teams from $r_{((j+1) \bmod (m/6))+1}$ in the given direction before it is being visited by them. After $m/6$ of these iterations, the teams $t_{1,j}$ for $j \leq m/6$ have played all their necessary matches against the small teams covered by $r_1, r_2, \dots, r_{m/6}$. In the same way, we let the teams $t_{1,i}$ visit the teams $s_{1,j}$ for $i, j \in \{m/6 + 1, \dots, m/3\}$. The only difference is that the $s_{1,j}$ are the first ones which travel and the $t_{1,i}$ travel then. In order to fill the gaps in the tours of the $s_{j,i}$, we take another $(2/3) \cdot m$ teams $t_{i,j}$ ($i \in \{2, 3\}, j \in \{1, m/3\}$) and let them repeat the tours, such that $t_{2,j}$ and $t_{3,j}$ play the same matches as $t_{1,j}$ just one and two days later, respectively. This way, we organize the games of $s_{i,j}$ against the teams $t_{i,j}$ such that the length of every tour is exactly three and all the tours through the set of small vertices are performed along the same trips. We repeat the whole procedure m^2 times and obtain a partition of W into the set T_1 of those teams that have already played all their matches against the set S_1 of teams which are covered by the trips r_j

for $j \leq m/6$ and the set T_2 of those teams that have played against the other teams $S_2 := S \setminus S_1$. We repeat the whole procedure with the teams of T_1 and T_2 changing their roles. This way, we obtain a schedule for the teams of S such that all of these teams are visited by the teams in W by using the given trips r_j . Additionally, every road trip or home stand has length 3 and, thus, the edges between S and W are used as few times as possible.

It remains to show that the games within the sets S and W can be performed in a feasible way. We first focus on the games between members of S . By $L_i := \cup_{j=1, \dots, m/3} s_{i,j}$ we denote the teams in S that are visited at the i th position of their corresponding trips. By construction, the teams in L_i play the teams of W on the days $2 \cdot i - 1$ to $2m^3 + 2i - 2$ and are available on the other days of the season (1 to $2m^3 + 2m - 1$). Figure 5 shows which days are available for L_1, L_2 , and L_3 for their games against other teams in S and which of them are already occupied by games against teams in W . Furthermore, the construction above forces some of the teams to have home or away games on certain days, which is also shown in Figure 5 by “H”s and “A”s, respectively.

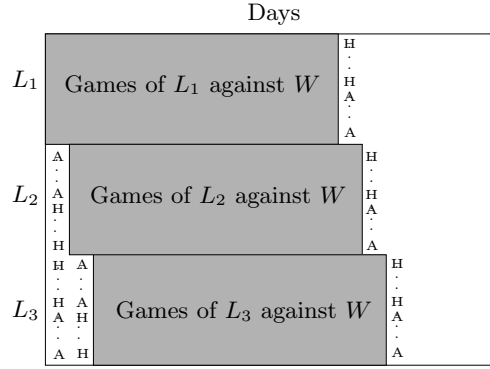
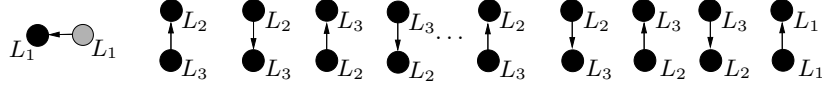
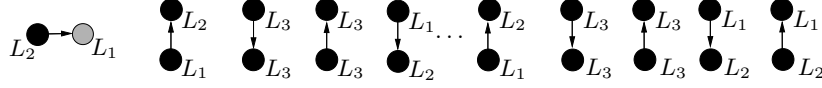


Fig. 5. Schedule of S after assignment of games against W

In order to schedule the matches between the teams in S , we apply the canonical tournament introduced by de Werra [12]. This way, we make sure that each team plays against every other team exactly once. This initial canonical schedule can be obtained by assigning the teams to the vertices of a special graph as displayed in Figure 6, thereby taking care that, for every i , we assign only teams from L_i to vertices with the respective subscript. The matches of the first time slot correspond to the pairs of vertices being adjacent to each other and the respective game always takes place at the head of the arc connecting them. The second day's matches can be obtained by changing the assignment of the teams to the gray vertices in the counterclockwise direction as shown in Figure 7. The schedules for the other time slots are derived analogously. Afterwards, the schedule is repeated with changed home field advantage.

Fig. 6. Games within S at first time slotFig. 7. Games within S at second time slot

The tournament we obtain this way has no road trip or home stand longer than three. Because of the way the teams from the L_i play against each other on the first two days of the tournament, it can fill the whitespace in the schedule displayed in Figure 5.

It remains to show how the teams from W play against each other. For every $w \in W$, let $d(w)$ denote the first day on which team w plays against a team from S and let $W_i := \{w \in W : (d(w) - 1 \bmod 3) + 1 = i\}$ for $i \in \{1, 2, 3\}$. Every W_i is partitioned into $2m^2$ groups of cardinality $m/6$ such that $d(w_1) = d(w_2)$ for every two members w_1, w_2 of the same group. For every W_i , we construct a schedule in which the members of different groups play against each other. As there will always be $m/3$ of these teams which are busy playing against S , we need to introduce two dummy-groups which represent S_1 and S_2 . We treat every group as a single team and apply the canonical tournament once again, only skipping the day at which the two dummy-groups representing S_1 and S_2 would meet.

The encounter of two groups being elements of W_i is realized in the following way: We organize the matches between members of two groups $h = \{h_1, h_2, \dots, h_l\}$ and $g = \{g_1, g_2, \dots, g_l\}$ in l rounds. The i th round contains the matches of team h_j against the team $g_{((j+i) \bmod l)+1}$ for all $i, j \in \{1, 2, \dots, l\}$ with the game taking place at the venue of h_j if and only if $i + j$ is even (which ensures that no road trip or home stand is too long). Afterwards, the games are repeated with changed home field advantage. We do not need to encounter between two groups in which one of them is a dummy-group, as these are matches between S and W which have already been taken care of above. The fundamental difference between these two different kinds of encounters is their length. The first kind mentioned takes $2 \cdot m/6$ days, whereas the latter one takes $2 \cdot m/2$ days. Therefore, there will always be $m/3$ days left which we can use to organize the games between different W_i . For this reason, we schedule the “free” days of the groups such that they coincide. As we skipped one “day” of the canonical schedule, there are still groups of size $m/3$ which have not played against each other. These games will be planned as canonical schedules as well and added at the end.

In order to organize the matches between the different W_i , we partition them into sets $W_{i,j}$ of equal size with $i \in \{1, 2, 3\}, j \in \{1, 2\}$. These encounters are again organized by computing a canonical schedule on the virtual teams $W_{i,j}$, skipping the day on which $W_{i,1}$ would play against $W_{i,2}$ for all i (which exists if the graph is properly initialized). The encounter of two groups represented by two different $W_{i,j}$'s is then realized in the same way as above, taking care that road trips or home stands are not too long. This completes the construction of our tournament.

Since the teams in t can visit each other at cost zero and we have seen in (1) that the distance traveled by the teams at the small vertices for visiting each other is at most $4m^2$, our tournament has total cost at most

$$\begin{aligned} & \underbrace{p \cdot ((16/3)M + 8)}_{t \text{ to small}} + \underbrace{2(m^3/3) \cdot (8pM + 6p)}_{\text{small to } t} + \underbrace{4m^2}_{\text{small to small}} \\ &= (16/3)M \cdot p \cdot (m^3 + 1) + 4m^2 \cdot (m \cdot p + 1) + 8p = \zeta, \end{aligned}$$

which finishes the first direction of the proof.

“ \Leftarrow ”: Now assume that φ is *not* satisfiable. Again consider the possible tours for the teams in t for visiting the teams at the small vertices. As we have seen above, every optimal set of such tours (even when not covering the vertices $w_{i,j}, z_{i,j}$) must use p tours with 2 edges in the stars from Figure 3. After doing so, it remains to cover p of the vertices $x_{i,j}/\bar{x}_{i,j}$, and all the vertices $w_{i,j}, z_{i,j}$ with exactly $(5/3)p$ tours. Covering the x -vertices by $(5/3)p$ tours results in a cost as least as large as in our set of tours constructed in the other direction of the proof. But since φ is not satisfiable now, it is no longer possible to choose the x -vertices contained in the star-tours such that they correspond to a truth assignment of the variables (choosing some $x_{i,j}$ means $x_i = \text{false}$ and choosing some $\bar{x}_{i,j}$ means $x_i = \text{true}$) while at the same time leaving at least one x -vertex connected to $w_{i,j}$ in G uncovered for all i, j . Thus, it is *not* possible in this case to extend any set of star-tours such that it also covers all vertices $w_{i,j}$ (and $y_{i,j}$) without increasing its cost by choosing edges not in G . Hence, the cost of every team in t for visiting the teams at the small vertices is at least one larger than in our schedule constructed above, which leads to an increase in total cost of at least m^3 . Since the only part of our schedule that was possibly suboptimal were the tours of total cost at most $4m^2$ used by the teams at the small vertices for visiting each other, this shows that there cannot exist a tournament of cost ζ for large enough values of m . This finishes the second direction of the proof. \square

References

1. Easton, K., Nemhauser, G., Trick, M.: The traveling tournament problem description and benchmarks. In: Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming (CP). Volume 2239 of LNCS. (2001) 580–584

2. Bhattacharyya, R.: A note on complexity of traveling tournament problem. *Optimization Online* (2009)
3. Benoist, T., Laburthe, L., Rottembourg, B.: Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problems. In: *Proceedings of the 3rd International Workshop on the Integration of AI and OR Techniques (CP-AI-OR)*. (2001) 15–26
4. Easton, K., Nemhauser, G., Trick, M.: 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 (PATAT)*. Volume 2740 of LNCS. (2003) 100–109
5. Anagnostopoulos, A., Michel, L., Van Hentenryck, P., Vergados, Y.: A simulated annealing approach to the travelling tournament problem. *Journal of Scheduling* **9**(2) (2006) 177–193
6. Miyashiro, R., Matsui, T., Imahori, S.: An approximation algorithm for the traveling tournament problem. In: *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT)*. (2008)
7. Yamaguchi, D., Imahori, S., Miyashiro, R., Matsui, T.: An improved approximation algorithm for the traveling tournament problem. In: *Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC)*. Volume 5878 of LNCS. (2009) 679–688
8. Kendall, G., Knust, S., Ribeiro, C., Urrutia, S.: Scheduling in sports: An annotated bibliography. *Computers and Operations Research* **37**(1) (2010) 1–19
9. Rasmussen, R., Trick, M.: Round robin scheduling - a survey. *European Journal of Operations Research* **188** (2008) 617–636
10. Garey, M.R., Johnson, D.S.: *Computers and Intractability (A Guide to the Theory of NP-Completeness)*. W.H. Freeman and Company, New York (1979)
11. Itai, A., Perl, Y., Shiloach, Y.: The complexity of finding maximum disjoint paths with length constraints. *Networks* **12**(3) (1982) 277–286
12. de Werra, D.: Scheduling in sports. In: *Studies on Graphs and Discrete Programming*. Volume 11 of *Annals of Discrete Mathematics*. North-Holland Publishing Company (1981) 381–395