European Journal of Operational Research 236 (2014) 592–600

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

# Improved bounds for the traveling umpire problem: A stronger formulation and a relax-and-fix heuristic ☆

Lucas de Oliveira [a], Cid C. de Souza [a], Tallys Yunes [b,*]

[a] Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil
[b] School of Business Administration, University of Miami, Coral Gables, FL, USA

ARTICLE INFO

*Article history:*
Received 29 August 2013
Accepted 11 December 2013
Available online 18 December 2013

*Keywords:*
OR in sports
Baseball
Heuristics
Integer programming
Relax-and-fix
Traveling umpire problem

ABSTRACT

Given a double round-robin tournament, the traveling umpire problem (TUP) consists of determining which games will be handled by each one of several umpire crews during the tournament. The objective is to minimize the total distance traveled by the umpires, while respecting constraints that include visiting every team at home, and not seeing a team or venue too often. We strengthen a known integer programming formulation for the TUP and use it to implement a relax-and-fix heuristic that improves the quality of 24 out of 25 best-known feasible solutions to instances in the TUP benchmark. We also improve all best-known lower bounds for those instances and, for the first time, provide lower bounds for instances with more than 16 teams.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction

The assignment of officials (referees, umpires, judges, etc.) to the games of a competition is an important and difficult problem studied in the area of sports scheduling. The specific constraints and objectives vary according to the sport and type of competition, of course, but they typically aim to satisfy a given set of fairness criteria while minimizing costs (e.g. wages or travel). Existing research ranges over many different sports, including baseball (Evans, Hebert, & Deckro, 1984; Evans, 1988; Trick & Yildiz, 2011; Trick & Yildiz, 2012; Trick, Yildiz, & Yunes, 2012), cricket (Wright, 1991), football (Yavuz, İnan, & Fığlalı, 2008), and tennis (Farmer, Smith, & Miller, 2007). For more comprehensive surveys of sports-related problems, we refer to Ernst, Jiang, Krishnamoorthy, Owens, and Sier (2004) and Kendall, Knust, Ribeiro, and Urrutia (2010).

We study the traveling umpire problem (TUP), which was first proposed by Trick and Yildiz (2007) as an abstract version of the real-life umpire scheduling problem faced by Major League Baseball. Despite excluding many details present in the real problem, the TUP successfully captures the most important features that make the problem very challenging to solve. This is evidenced by the fact that many small instances remain unsolved in the official TUP benchmark: Trick (2013).

Given a double round-robin tournament with $2n$ teams (each team plays against each other team twice, once at home and once on the road, over exactly $4n-2$ rounds), the distances between the home venues of each pair of teams, and two integers $0 \leqslant d_1 < n$ and $0 \leqslant d_2 < \lfloor \frac{n}{2} \rfloor$, a solution to the TUP is an assignment of $n$ umpire crews (umpires, for short) that satisfies the following constraints:

(i) In each round, each umpire is assigned to exactly one game and each game must be assigned to an umpire;
(ii) Each umpire visits every team at home at least once;
(iii) Each umpire visits any given venue at most once in any sequence of $n-d_1$ consecutive games;
(iv) Each umpire sees any given team at most once in any sequence of $\lfloor \frac{n}{2} \rfloor - d_2$ consecutive games.

The objective is to find a feasible solution that minimizes the total distance traveled by the umpires over the entire tournament.

When $d_1 = d_2 = 0$, TUP instances tend to be more difficult to solve because constraints (iii) and (iv) become stricter. We refer to these instances as *hard instances*. To allow for a wider range in the degree of difficulty, the TUP benchmark also includes instances with $d_1 + d_2 > 0$, to which we refer as *relaxed instances*.

☆ This research was supported by Grant 01-P-01965-2012 from *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (Brazil) and Grants 142278/2013-0, 477692/2012-5 and 302804/2010-2 from *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (Brazil).

* Corresponding author. Address: School of Business Administration, University of Miami, 5250 University Drive, Room KE405, Coral Gables, FL 33146, USA. Tel.: +1 305 284 5107; fax: +1 305 284 2321.

*E-mail addresses:* lucas.oliveira@ic.unicamp.br (L. de Oliveira), cid@ic.unicamp.br (C.C. de Souza), tallys@miami.edu (T. Yunes).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.12.019

Our main contributions are: (1) we strengthen a known integer programming formulation for the TUP and use it to implement a relax-and-fix heuristic that improves the quality of 24 out of 25 best-known feasible solutions to instances in the TUP benchmark; and (2) using our stronger formulation, we improve all best-known lower bounds for those instances and, for the first time, provide lower bounds for instances with more than 16 teams.

Before explaining our approach in detail, we review the existing exact and heuristic methods for solving the TUP.

## 2. Previous work

Trick and Yildiz (2007) present exact integer programming (IP) and constraint programming (CP) models for the TUP and test them on benchmark instances ranging from 4 to 16 teams, as well as on a 30-team instance. These same models are also used in Trick and Yildiz (2011); Trick and Yildiz (2012); and Trick et al. (2012), but in Trick and Yildiz (2011) and Trick and Yildiz (2012) the performance of the IP model is improved by a better choice of solver parameters (execution times were limited to three hours). Their IP model finds optimal solutions to all instances with at most 10 teams. The CP model finds optimal solutions to all instances with at most 8 teams, and also to one of the four 10-team instances. In addition, it manages to prove that the 12-team instance is infeasible; a conclusion that was not obtained by the IP model within the allowed computation time. When it comes to hard instances with 14 teams, both IP and CP find feasible solutions to all four instances, with the CP model beating the IP model in terms of solution quality in three of the four cases. Neither the IP nor the CP models managed to find any feasible solutions to hard instances with more than 14 teams. When run on relaxed instances, the CP model finds feasible solutions to all eight 14-team and all twelve 16-team instances. The IP model finds feasible solutions to 17 of these 20 instances (three 16-team instances had no solution after 3 hours), but all 17 solutions are better than their counterpart solutions found by the CP model. Neither method managed to find provably optimal solutions to any of the relaxed instances with more than 10 teams.

On the heuristic side, we focus on hard and relaxed instances with at least 14 teams because none of them have known optimal solutions (29 instances in total; 25 with known feasible solutions). Trick et al. (2012) describe a greedy matching heuristic (GMH) and a two-exchange local search that are combined to build a simulated annealing (SA) heuristic. The SA heuristic does reasonably well on the real-life Major League Baseball problem of 2006, but it does not perform so well on the TUP instances. Trick and Yildiz (2011) incorporate the GMH into a large neighborhood search guided by Benders cuts that help repair the solution being built when the heuristic gets stuck. This improved GMH, which they call GBNS, finds feasible solutions to 23 out of 25 instances with previously known solutions, with 16 of those solutions being improvements over the best results at the time. In a follow-up paper, Trick and Yildiz (2012) propose a genetic algorithm (GA) with a crossover operator that uses a matching scheme to recombine the individuals of a population. Their GA further improves the quality of 14 instances with respect to the GBNS results. As of August, 2013, the results published in Trick (2013) indicate that the four best-known solutions to the 14-team hard instances were obtained by Wauters (2013). Table 1 shows how many of the 25 best-known solutions to date have been found by each of the most successful methods described above. The GA currently owns the majority of best-known solutions (13 out of 25), including the best solution to the 30-team instance.

The remainder of this paper is organized as follows. In Section 3, we present the IP formulation of Trick and Yildiz (2011) and show how it can be strengthened. We describe our relax-and-fix heuristic in Section 4, and provide computational results in Section 5. We conclude the paper and discuss future research directions in Section 6.

## 3. IP formulations

For simplicity, we use letters $i$ and $j$ to refer to teams $i$ and $j$, as well as their respective home venues. In addition, we use letters $u$ and $s$ to refer to an umpire and a round in the tournament, respectively.

### 3.1. The original formulation

The IP formulation used by Trick and Yildiz (2011); Trick and Yildiz (2012); and Trick et al. (2012) starts with the following input data:

- Set of umpires $U = \{1, \dots, n\}$;
- Set of teams $T = \{1, \dots, 2n\}$;
- Set of rounds $S = \{1, \dots, 4n-2\}$;
- $\text{OPP}[s,i] = \begin{cases} j & \text{if } i \text{ plays against } j \text{ at venue } i \text{ in round } s \\ -j & \text{if } i \text{ plays against } j \text{ at venue } j \text{ in round } s; \end{cases}$
- $d_{ij}$ = distance in miles between venues $i$ and $j$;
- $CV_s = \{s, \dots, s+n-d_1-1\}$ for any given round $s \in \{1, \dots, 4n-2-(n-d_1-1)\}$;
- $CT_s = \{s, \dots, s+\lfloor\frac{n}{2}\rfloor - d_2 - 1\}$ for any given round $s \in \{1, \dots, 4n-2-(\lfloor\frac{n}{2}\rfloor - d_2 - 1)\}$.

The decision variables are:

- $x_{isu} = \begin{cases} 1 & \text{if the game at venue } i \text{ inround } s \text{ isassigned} \\ & \text{toumpire } u \\ 0 & \text{otherwise;} \end{cases}$
- $z_{ijsu} = \{1$ if umpire $u$ is at venue $i$ in round $s$ and travels to venue $j$ in round $s+1$0 otherwise.

We are now ready to state the formulation.

$$\min \sum_{i\in T}\sum_{j\in T}\sum_{u\in U}\sum_{s\in S: s<|S|} d_{ij} z_{ijsu} \tag{1}$$

$$\sum_{u\in U} x_{isu} = 1, \quad \forall\, i \in T, s \in S : \text{OPP}[s,i] > 0, \tag{2}$$

$$\sum_{i\in T:\text{OPP}[s,i]>0} x_{isu} = 1, \quad \forall\, s \in S, u \in U, \tag{3}$$

$$\sum_{s\in S:\text{OPP}[s,i]>0} x_{isu} \geqslant 1, \quad \forall\, i \in T, u \in U, \tag{4}$$

$$\sum_{c\in CV_s:\text{OPP}[c,i]>0} x_{icu} \leqslant 1, \quad \begin{array}{l}\forall\, i \in T, u \in U, s \in S : \\ s \leqslant |S| - (n-d_1-1),\end{array} \tag{5}$$

$$\sum_{c\in CT_s}\left(x_{icu} + \sum_{j\in T:\text{OPP}[c,j]=i} x_{jcu}\right) \leqslant 1, \quad \begin{array}{l}\forall\, i \in T, u \in U, s \in S : \\ s \leqslant |S| - (\lfloor\frac{n}{2}\rfloor - d_2 - 1),\end{array} \tag{6}$$

$$x_{isu} + x_{j(s+1)u} - z_{ijsu} \leqslant 1, \quad \forall\, i,j \in T, u \in U, s \in S : s < |S|, \tag{7}$$

$$x_{isu} \in \{0,1\}, \quad \forall\, i \in T, u \in U, s \in S, \tag{8}$$

$$z_{ijsu} \in \{0,1\}, \quad \forall\, i,j \in T, u \in U, s \in S : s < |S|. \tag{9}$$

The objective function (1) minimizes the total distance traveled by the umpires. Constraints (2) and (3) state that each game is refereed by an umpire, and each umpire is assigned to a game, respectively. TUP constraints (ii), (iii), and (iv) from Section 1 are modeled by (4)–(6), respectively. Finally, (7) ensures that game ($x$) and travel ($z$) assignments are consistent.

Trick and Yildiz (2011) improve the above formulation by including the following constraints:

**Table 1**
Number of best-known solutions found by existing solution methods according to the TUP benchmark web site.

| Solution method | Instance type/# teams | | | |
|---|---|---|---|---|
| | $d_1 = d_2 = 0$ | | $d_1 + d_2 > 0$ | |
| | 14 | 16 | 14 | 16 |
| IP of Trick and Yildiz (2011) | | | 1 | 2 |
| GBNS of Trick and Yildiz (2011) | | | | 5 |
| GA of Trick and Yildiz (2012) | | | 7 | 6 |
| Wauters (2013b) | 4 | | | |

$$x_{isu} = 0, \quad \forall\, i \in T, u \in U, s \in S : \mathrm{OPP}[s,i] < 0, \tag{10}$$
$$z_{ijsu} - x_{isu} \leqslant 0, \quad \forall\, i,j \in T, u \in U, s \in S : s <| S |, \tag{11}$$
$$z_{ijsu} - x_{j(s+1)u} \leqslant 0, \quad \forall\, i,j \in T, u \in U, s \in S : s <| S |, \tag{12}$$
$$\sum_{j \in T} z_{jisu} - \sum_{j \in T} z_{ij(s+1)u} = 0, \quad \forall\, i \in T, u \in U, s \in S : s <| S | -1, \tag{13}$$
$$\sum_{i \in T}\sum_{j \in T} z_{ijsu} = 1, \quad \forall\, u \in U, s \in S : s <| S |\,. \tag{14}$$

Constraint (10) forbids the assignment of an umpire to a venue where no game is played in the given round. Constraints (11) and (12) allow an umpire to move from or to a venue in round $s$ and $s+1$, respectively, only if the umpire is assigned to the venue in those rounds. The flow conservation constraint (13) states that if umpire $u$ is (is not) at venue $i$ in round $s+1$, his trip from round $s+1$ to round $s+2$ must (must not) start at venue $i$. Constraint (14) forces every umpire to move from one round to the next.

Formulation (1)–(14) suffers from symmetry because, given a solution, $n!$ equivalent solutions can be obtained by permuting the umpires. To overcome this problem, Yildiz (2008) fixes the games refereed by each umpire in a given round $k$, which is chosen arbitrarily in the interval $[1, 4n-2]$. To accomplish that, we enforce (15) for a given set $K$ of tuples that assign each umpire to a different venue where a game takes place in round $k$:

$$x_{iku} = 1, \quad \forall\, (i,u) \in K. \tag{15}$$

We refer to round $k$ as the *symmetry breaking round*, and to formulation (1)–(15) as $\mathcal{F}_1$. The linear relaxation of $\mathcal{F}_1$, in which we replace the integrality constraints 8–9 with $0 \leqslant x_{isu} \leqslant 1$ and $0 \leqslant z_{ijsu} \leqslant 1$, is denoted by $\mathcal{F}_1^R$.

### 3.2. A stronger formulation

Based on $\mathcal{F}_1$, we propose a stronger formulation for the TUP that includes the following valid identities:

$$x_{i1u} = \sum_{j \in T} z_{ij1u} \quad \forall\, i \in T, u \in U, \tag{16}$$
$$x_{isu} = \sum_{j \in T} z_{ji(s-1)u} \quad \forall\, i \in T, u \in U, s \in S : s > 1. \tag{17}$$

Constraint (16) ensures that an umpire $u$ is assigned to a venue $i$ in the first round if, and only if, $u$ travels away from $i$ in this round. Analogously, (17) enforces that an umpire $u$ is assigned to a venue $i$ in round $s > 1$ if, and only if, $u$ travels to $i$ between rounds $s-1$ and $s$.

Because (16)–(17) are equalities, variable $x$ can be eliminated from $\mathcal{F}_1$ by replacing each of its occurrences with the proper summation over $z$. Let $\mathcal{F}_2$ be the formulation that results from $\mathcal{F}_1$ after *both* eliminating $x$ and removing (3), (7), (11), (12), and (14). We refer to the linear relaxation of $\mathcal{F}_2$ as $\mathcal{F}_2^R$.

**Proposition 1.** *$\mathcal{F}_2$ is a valid IP formulation for the TUP.*

**Proof.** It suffices to show that (3), (7), (11), (12), and (14) are implied by the conjunction of (16) and (17) with the remaining constraints of $\mathcal{F}_1$. For simplicity, and without loss of generality, we will show the implications before eliminating $x$ from the model.

First, by combining (13) and (16), (17) we obtain the following identity:

$$x_{isu} = \sum_{j \in T} z_{ijsu}, \quad \forall\, i \in T, u \in U, s \in S : s <| S |\,. \tag{18}$$

To see that (18) implies (11), note that $\sum_{j \in T} z_{ijsu} \geqslant z_{ihsu}$ for any given $h \in T$. The proof that (17) implies (12) is analogous.

We now show that (3) and (14) are implied constraints. Because of (13), we can write

$$\sum_{i \in T}\sum_{j \in T} z_{ij1u} = \ldots = \sum_{i \in T}\sum_{j \in T} z_{ijku} = \ldots = \sum_{i \in T}\sum_{j \in T} z_{ij(|S|-1)u}, \quad \forall\, u \in U, \tag{19}$$

and, by replacing the inner summations in (19) with the appropriate $x$ variables according to (18), we obtain the equivalent equalities in (20), except for the last one:

$$\sum_{i \in T} x_{i1u} = \ldots = \sum_{i \in T} x_{iku} = \ldots = \sum_{i \in T} x_{i(|S|-1)u} = \sum_{i \in T} x_{i(|S|)u}, \quad \forall\, u \in U. \tag{20}$$

To obtain the last equality in (20) we replace the rightmost inner summation in (19) with the appropriate $x$ variable according to (17).

Because of (15), $\sum_{i \in T} x_{iku} = 1$. Thus, all the expressions in (19) and (20) are equal to 1. Since the summations in (19) and (20) are the same as those in (14) and (3), respectively, the latter are satisfied.

Finally, we show that (7) is an implied constraint. For any $u \in U, s \in S, s < |S|$, we have seen that combining (15), (19), and (20) yields

$$\sum_{i \in T}\sum_{j \in T} z_{ijsu} = 1.$$

Hence, for a given pair of venues $i,j$ in the double-summation above, we have

$$\sum_{h \in T : h \neq j} z_{ihsu} + \sum_{h \in T : h \neq i} z_{hjsu} + z_{ijsu} \leqslant 1,$$

which is equivalent to

$$\sum_{h \in T} z_{ihsu} + \sum_{h \in T} z_{hjsu} - z_{ijsu} \leqslant 1. \tag{21}$$

Because of (18), we can substitute $x_{isu}$ for $\sum_{h \in T} z_{ihsu}$ in (21). Similarly, (17) allows us to substitute $x_{j(s+1)u}$ for $\sum_{h \in T} z_{hjsu}$ in (21), turning it into (7). □

We are now ready to show that $\mathcal{F}_2$ is stronger than $\mathcal{F}_1$.

**Proposition 2.** *The lower bound provided by $\mathcal{F}_2^R$ is greater than or equal to the lower bound provided by $\mathcal{F}_1^R$. Moreover, the former bound can be strictly greater than the latter.*

**Proof.** From the proof of Proposition 1, it is clear that the lower bound provided by $\mathcal{F}_2^R$ cannot be worse than the lower bound provided by $\mathcal{F}_1^R$. Therefore, it suffices to show an example in which the former bound is strictly greater than the latter. Consider a TUP instance with four teams (represented by a table of games in each round) as shown in Fig. 1. The first team in each pair (game) is the home team. Suppose that teams 1, 2, and 3 have their venues close to each other (say, 1 mile away), and the venue of team 4 is further away from the others (say, 10 miles away). In this scenario, an optimal solution to $\mathcal{F}_1^R$ with $d_1 = d_2 = 0, k = 1$, and $K = \{(1,1),(3,2)\}$ corresponds to assigning value 1 to variables $x_{111}$ and $x_{312}$, value 0 to variables $x_{311}$ and $x_{112}$, value 0.5 to all other $x_{isu}$ variables where $i \in T, s \in S, u \in U$, and $OPP[i,s] > 0$, value 0.5 to variables $z$ of umpire 1 associated with the trips given by the simple solid and dashed lines in Fig. 1, and value 0.5 to variables $z$ of umpire 2 associated with the trips given by the simple and double solid lines in Fig. 1. All remaining variables are equal to zero. The optimal value is 25, which is a very weak lower bound because the value of an optimal integer solution to this instance is 55.

Now, observe that (17) is not satisfied by the previous solution because variables $x_{441}, x_{442}, x_{461}$, and $x_{462}$ have value 0.5 and all the $z$ variables arriving to or leaving from the venue of team 4 in rounds 4 and 6 have value 0. Hence, this fractional solution becomes infeasible once (17) is added to $\mathcal{F}_1^R$. In fact, the new optimal solution found after the inclusion of (16) and (17) has value 52, which is much closer to the integer optimum. □

It is possible to strengthen $\mathcal{F}_2^R$ further by fixing the following variables:

$$z_{iisu} = 0, \quad \forall\, i \in T, u \in U, s \in S : s < |S|, \text{ if } d_1 < n-1, \tag{22}$$

$$z_{ijsu} = 0, \quad \begin{array}{l}\forall\, i \neq j \in T, u \in U, s \in S : s < |S|, \text{ if } d_2 < \lfloor \frac{n}{2} \rfloor - 1 \text{ and} \\ \text{either } \mathrm{OPP}[s,i] = \mathrm{OPP}[s+1,j] \text{ or } \mathrm{OPP}[s,i] = j \text{ or} \\ \mathrm{OPP}[s+1,j] = i.\end{array} \tag{23}$$

Note that (22) and (23) are valid because they forbid the assignment of positive values to $z$ variables that would violate TUP constraints (iii) and (iv) (no umpire can stay at a venue, or follow a team) when $d_1$ and $d_2$ are strictly less than their maximum values. Let $\mathcal{F}_{2+}$ and $\mathcal{F}_{2+}^R$ be, respectively, the formulations obtained from $\mathcal{F}_2$ and $\mathcal{F}_2^R$ after the inclusion of (22) and (23). In the example of Fig. 1, the optimal solution obtained by $\mathcal{F}_1^R$ (see proof of Proposition 2) violates (22) multiple times. For example, it sets $z_{2221} = z_{2222} = z_{2231} = z_{2232} = z_{3351} = z_{3352} = 0.5$. Solving $\mathcal{F}_{2+}^R$ for that example yields an integer (optimal) solution of value 55. Recall that the lower bounds produced by $\mathcal{F}_1^R$ and $\mathcal{F}_2^R$ for that example are 25 and 52, respectively.

## 4. A relax-and-fix heuristic

A relax-and-fix heuristic iteratively solves relaxations of an IP model and progressively fixes variables until a feasible solution is found (Wolsey, 1998, Section 12.5). Our relax-and-fix heuristic is based on $\mathcal{F}_{2+}$. It receives as input a TUP instance and an integer $1 \leqslant b \leqslant 4n-3$. The parameter $b$ defines the size of a window of consecutive rounds whose variables will be binary in the relaxed model solved at each iteration of the algorithm. Starting with formulation $\mathcal{F}_{2+}$ with the symmetry breaking round $k = 1$, we modify it so that only the variables in the first $b$ rounds are binary. This relaxed model is solved and, if it is infeasible, the algorithm stops. Otherwise, the variables in the first round are fixed to their values in the best solution found within some stopping criteria. In the next iteration, variables in round $b+1$ are also made binary and the resulting model is solved. As before, we stop in case of infeasibility, and fix the variables in the second round to their best values otherwise. These steps are repeated until either all variables are fixed, or the model becomes infeasible. The pseudocode in Fig. 2 includes more specific details, which we discuss next.

At iteration $t$, the heuristic solves a relaxed $\mathcal{F}_{2+}$ model in which the $z$ variables in rounds $s \leqslant t-1, t \leqslant s < t+b$, and $s \geqslant t+b$ are fixed, binary, and continuous, respectively. If this model is feasible, the variables in round $t$ are fixed to their values in the best solution found after exploring no more than a given number of search nodes (see Section 5.3 for details). Because intermediate solutions found in line 8 of the pseudocode are not necessarily optimal for the last model solved, the heuristic may continue to execute until iteration $t = 4n-3$, when all variables in the model are fixed.

To keep running times under control, we use two different strategies to solve the relaxations of $\mathcal{F}_{2+}$ in each iteration of the relax-and-fix heuristic. For instances with at most 18 teams, we solve the models with all variables and constraints, as described above. For instances with more than 18 teams, in each iteration $t$ of the heuristic we solve a relaxation of $\mathcal{F}_{2+}$ that only includes the variables and constraints relative to rounds no greater than $\min(\lfloor \frac{2}{5}(4n-2) \rfloor + t - 1, 4n-2)$. (Recall that, in iteration $t$, all variables relative to the $t-1$ first rounds are fixed.) This is equivalent to redefining the set $S$ at each iteration to be $S = \{1, \ldots, \min(\lfloor \frac{2}{5}(4n-2) \rfloor + t - 1, 4n-2)\}$. The value $\lfloor \frac{2}{5}(4n-2) \rfloor$ was chosen experimentally with the goal of accelerating the heuristic, while still allowing the model to look at enough rounds ahead of round $t$ to be able to find good solutions. To properly disregard future rounds while solving relaxations of $\mathcal{F}_{2+}$, it is necessary to deal with (4) in a different way. In early iterations, the number of rounds considered by the relaxed $\mathcal{F}_{2+}$ models is insufficient to satisfy the constraint. If we disregard (4) entirely for many iterations, however, the variables that get fixed may make it impossible to satisfy (4) later on. Therefore, we introduce (4) gradually as follows. We disregard it during iterations $t \leq n$. In iterations $n+1, n+2, \ldots, 2n-1$ we enforce (4) only for umpires whose indices are less than or equal to $1, 2, \ldots, n-1$, respectively, and in iterations $t > 2n-1$ we enforce (4) for all umpires.

Notice that our heuristic begins with a fixed assignment of umpires for the first round ($k = 1$) and, at each iteration $t$, determines the umpire trips between rounds $t$ and $t+1$. This resembles the GMH of Trick et al. (2012). The difference stems from the fact that our relax-and-fix heuristic schedules the umpires in each round while taking into account, in a relaxed way, the subsequent rounds of the tournament. In addition, we deal with the trip distances directly, rather than using a modified cost function to schedule each round. Finally, because our method is not guaranteed to find a solution every time, we could use a backtracking scheme similar to what is done with the GMH. In practice, however, we almost always find solutions to the instances that are known to be feasible.

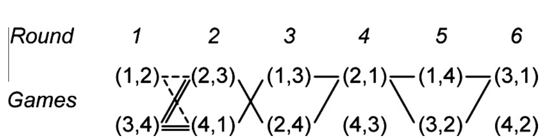


**Fig. 1.** A TUP instance. Dashed, simple and double solid lines represent umpire trips.

```
1: procedure RELAX-AND-FIX(TUP instance I, b)
2:     M ← F_{2+} model with k = 1 for instance I;
3:     t ← 1;
4:     while t ≤ 4n − 3 do
5:         For all i, j ∈ T, u ∈ U, s ∈ S, t ≤ s < t + b, let z_{ijsu} ∈ {0,1} in M;
6:         For all i, j ∈ T, u ∈ U, s ∈ S, s ≥ t + b, let 0 ≤ z_{ijsu} ≤ 1 in M;
7:         Solve M;
8:         if a solution z̄_{ijsu} is found within the allowable # of search nodes then
9:             Fix variables z_{ijsu} = z̄_{ijsu} for all i, j ∈ T, u ∈ U, s = t;
10:            t ← t + 1;
11:        else
12:            return no solution;
13:        end if
14:    end while
15:    return solution found;
16: end procedure
```

**Fig. 2.** Pseudocode of the relax-and-fix heuristic.

Therefore, to keep our implementation simple, we decided not to use a backtracking step.

## 5. Computational results

In this section, we experimentally evaluate the IP models and the relax-and-fix heuristic presented earlier. We consider instances from the TUP benchmark (Trick, 2013) ranging from 4 to 32 teams. Instance names start with the number of teams, optionally followed by a letter indicating that the distance matrix is different, but the underlying tournament is not. Experiments were performed on a machine running Linux Ubuntu 12.04.1 and equipped with an zi7-2600, 3.40 GHz processor, and 8 GB of RAM. Our code is written in C++, and we use ILOG CPLEX's callable library version 12.5.0.1 to solve linear programming and integer programming models.

### 5.1. IP results on hard instances: Lower and upper bounds

Before solving the IP models, we analyze the strength of our lower bounds. We focus first on hard instances ($d_1 = d_2 = 0$), and fix the symmetry breaking round $k = 1$. Table 2 shows the lower bounds obtained by solving the linear relaxations of the original IP model ($\mathcal{F}_1^R$) and of our stronger formulations ($\mathcal{F}_2^R$) and ($\mathcal{F}_{2+}^R$). We also include the solution time and the optimal distances for instances with known optimal solutions. Note that, except for the smallest instance (4 teams), $\mathcal{F}_2^R$ and $\mathcal{F}_{2+}^R$ always provide better lower bounds than $\mathcal{F}_1^R$. In particular, $\mathcal{F}_{2+}^R$ provides significantly better bounds (up to 340% better) that are reasonably close to the known optimal solutions, which attests the importance of (22) and (23). In addition, $\mathcal{F}_2^R$ and $\mathcal{F}_{2+}^R$ solve between 2 and 8 times faster than $\mathcal{F}_1^R$ because they are more compact (fewer variables and constraints), with $\mathcal{F}_{2+}^R$ having a slight advantage over $\mathcal{F}_2^R$.

Our experiments show that the choice of the symmetry breaking round $k$ can significantly affect the value of the lower bound provided by the linear relaxations. However, it is difficult to determine a priori what value of $k$ leads to the best results for the integer models. Therefore, we solve the IP models with three different values of $k$ : 1 (first round), $2n - 1$ (middle round), and the value $k^*$ that produces the best lower bound. The models adjusted with each chosen value of $k$ are denoted as before but followed by the suffixes "-F" (first-round $k$), "-M" (middle-round $k$) and "-BB" (best-bound $k$), respectively.

We now switch from root-node lower bounds to solving the IP models, while still focusing on hard instances. We limit execution times to 3 hours and limit the number of CPU threads to one. Table 3 shows the optimal distances and execution times for hard instances with 12 or fewer teams. In general, $\mathcal{F}_{2+}$ exhibits better performance than $\mathcal{F}_1$. For a given formulation, breaking symmetry in the middle round often speeds up the optimization, whereas breaking symmetry in the first round tends be a bad idea. Therefore, we disregard $\mathcal{F}_1$-F and $\mathcal{F}_{2+}$-F in subsequent experiments.

Table 4 summarizes the results for hard instances with at least 14 teams, which have no known optimal solutions. Calculating $k^*$ for $\mathcal{F}_1$-BB on instances with at least 26 teams, and for $\mathcal{F}_{2+}$-BB on instance 32 takes longer than 3 hours. In these cases, we report the best lower bound found within 3 hours, which is marked with an "$*$", and do not execute CPLEX's branch-and-cut. The lower bounds in Table 4 contain unexpected results for $\mathcal{F}_1$. Although $\mathcal{F}_1^R$ does not perform well in Table 2, it appears that the root-node presolve and strengthening routines of CPLEX are able to improve it, enabling $\mathcal{F}_1$ to achieve competitive lower bounds in 3 hours. For example, the best lower bounds for instances with 16 and 18 teams are given by $\mathcal{F}_1$-BB. On the other hand, this model yields very poor lower bounds for instances with more than 24 teams within the time limit. The $\mathcal{F}_{2+}$ models achieve the best lower bounds on instances with 14 teams, or with more than 18 teams, with the latter being significantly better than those provided by the $\mathcal{F}_1$ models. Overall, $\mathcal{F}_{2+}$-BB obtains 8 out of the 16 best lower bounds,

**Table 2**
Lower bounds from the linear relaxations of different IP models of hard instances ($d_1 = d_2 = 0$) with $k = 1$.

| Instance | Optimum | Lower bound | | | Time (seconds) | | |
|---|---|---|---|---|---|---|---|
| | | $\mathcal{F}_1^R$ | $\mathcal{F}_2^R$ | $\mathcal{F}_{2+}^R$ | $\mathcal{F}_1^R$ | $\mathcal{F}_2^R$ | $\mathcal{F}_{2+}^R$ |
| 4 | 5176 | 5176.0 | 5176.0 | 5176.0 | 0.0 | 0.0 | 0.0 |
| 6 | 14,077 | 5874.3 | 7971.0 | 14077.0 | 0.0 | 0.0 | 0.0 |
| 6A | 15,457 | 7351.0 | 9228.0 | 13672.3 | 0.0 | 0.0 | 0.0 |
| 6B | 16,716 | 6541.0 | 9766.0 | 15786.3 | 0.0 | 0.0 | 0.0 |
| 6C | 14,396 | 5097.7 | 6717.3 | 14396.0 | 0.0 | 0.0 | 0.0 |
| 8 | 34,311 | 11836.7 | 14787.3 | 33723.2 | 0.1 | 0.0 | 0.0 |
| 8A | 31,490 | 11659.4 | 13792.4 | 30193.9 | 0.0 | 0.0 | 0.0 |
| 8B | 32,731 | 11057.2 | 15408.7 | 31724.4 | 0.0 | 0.0 | 0.0 |
| 8C | 29,879 | 10717.6 | 12532.7 | 27718.3 | 0.0 | 0.0 | 0.0 |
| 10 | 48,942 | 16781.1 | 20026.1 | 48040.1 | 0.2 | 0.1 | 0.1 |
| 10A | 46,551 | 13261.6 | 14991.3 | 44909.8 | 0.2 | 0.1 | 0.1 |
| 10B | 45,609 | 13832.1 | 17358.5 | 44238.7 | 0.2 | 0.1 | 0.1 |
| 10C | 43,149 | 14922.3 | 16594.7 | 39618.2 | 0.2 | 0.1 | 0.1 |
| 12 | | 26993.5 | 39722.3 | 82753.4 | 0.6 | 0.3 | 0.3 |
| 14 | | 43664.2 | 68706.4 | 140180.0 | 1.7 | 0.8 | 0.9 |
| 14A | | 42484.1 | 67680.4 | 132063.0 | 1.7 | 0.9 | 0.8 |
| 14B | | 41587.2 | 63116.8 | 129671.0 | 1.8 | 0.8 | 0.8 |
| 14C | | 45634.8 | 68155.2 | 125719.0 | 2.0 | 0.9 | 0.8 |
| 16 | | 46133.6 | 65830.9 | 131264.0 | 6.1 | 2.3 | 2.0 |
| 16A | | 51454.1 | 76574.5 | 145901.0 | 5.5 | 2.3 | 2.0 |
| 16B | | 48635.1 | 86315.4 | 143592.0 | 6.1 | 2.6 | 2.0 |
| 16C | | 36402.5 | 81699.4 | 144402.0 | 4.8 | 2.1 | 1.9 |
| 18 | | 49031.3 | 88031.7 | 152548.0 | 14.2 | 4.2 | 4.0 |
| 20 | | 47183.0 | 92467.9 | 183820.0 | 40.4 | 9.3 | 8.0 |
| 22 | | 55629.1 | 109778.0 | 209092.0 | 73.4 | 21.2 | 15.2 |
| 24 | | 56630.0 | 119211.0 | 226090.0 | 136.9 | 35.2 | 28.9 |
| 26 | | 59919.4 | 129532.0 | 266366.0 | 302.3 | 69.3 | 47.1 |
| 28 | | 72861.6 | 161630.0 | 302138.0 | 463.9 | 100.3 | 85.9 |
| 30 | | 83710.2 | 166316.0 | 338267.0 | 1169.8 | 170.7 | 143.6 |
| 32 | | 97058.9 | 186105.0 | 371968.0 | 1786.5 | 322.0 | 227.8 |

**Table 3**
Optimal results obtained by IP formulations on hard instances with at most 12 teams ($d_1 = d_2 = 0$).

| Instance | Optimum | Time (seconds) | | | | | |
|---|---|---|---|---|---|---|---|
| | | $\mathcal{F}_1$-F | $\mathcal{F}_1$-M | $\mathcal{F}_1$-BB | $\mathcal{F}_{2+}$-F | $\mathcal{F}_{2+}$-M | $\mathcal{F}_{2+}$-BB |
| 4 | 5176 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6 | 14,077 | 0.02 | 0.02 | 0.02 | 0.02 | 0.01 | 0.02 |
| 6A | 15,457 | 0.05 | 0.02 | 0.03 | 0.02 | 0.01 | 0.01 |
| 6B | 16,716 | 0.03 | 0.03 | 0.04 | 0.02 | 0.02 | 0.02 |
| 6C | 14,396 | 0.02 | 0.02 | 0.03 | 0.01 | 0.01 | 0.02 |
| 8 | 34,311 | 0.53 | 0.06 | 0.15 | 0.38 | 0.03 | 0.03 |
| 8A | 31,490 | 0.30 | 0.06 | 0.05 | 0.24 | 0.03 | 0.02 |
| 8B | 32,731 | 0.30 | 0.06 | 0.06 | 0.12 | 0.02 | 0.02 |
| 8C | 29,879 | 0.65 | 0.06 | 0.04 | 0.53 | 0.03 | 0.02 |
| 10 | 48,942 | 38.10 | 1.68 | 1.64 | 14.56 | 1.17 | 1.87 |
| 10A | 46,551 | 52.05 | 4.77 | 5.48 | 21.25 | 1.00 | 4.27 |
| 10B | 45,609 | 25.52 | 2.32 | 3.59 | 5.67 | 0.73 | 0.73 |
| 10C | 43,149 | 335.33 | 9.19 | 13.54 | 34.49 | 2.78 | 1.92 |
| 12 | Infeasible | 3172.34 | 101.10 | 857.48 | 2556.69 | 71.47 | 627.42 |

**Table 4**
Best results for the IP models on hard instances with at least 14 teams ($d_1 = d_2 = 0$). Execution times limited to 3 hour. Best values appear in bold.

| Instance | Best lower bound after 3 hour | | | | Best solution after 3 hour | | | |
|---|---|---|---|---|---|---|---|---|
| | $\mathcal{F}_1$-M | $\mathcal{F}_1$-BB | $\mathcal{F}_{2+}$-M | $\mathcal{F}_{2+}$-BB | $\mathcal{F}_1$-M | $\mathcal{F}_1$-BB | $\mathcal{F}_{2+}$-M | $\mathcal{F}_{2+}$-BB |
| 14 | 149,636 | 149,963 | 150,045 | **150,871** | 179,343 | 177,139 | 175,524 | **174,715** |
| 14A | 142,394 | 142,759 | **143,517** | 141,308 | 166,712 | 172,367 | **165,968** | 171,961 |
| 14B | 141,795 | 141,608 | 141,645 | **142,614** | 172,828 | 176,709 | **168,659** | 170,804 |
| 14C | 141,196 | 141,148 | **141,268** | 138,601 | 168,037 | 168,787 | **164,512** | 167,898 |
| 16 | 148,059 | **151,748** | 147,906 | 150,523 | | | | |
| 16A | 161,087 | **166,626** | 160,831 | 166,101 | | | | |
| 16B | 158,322 | **162,251** | 159,324 | 161,882 | | | | |
| 16C | 162,640 | **165,431** | 162,505 | 164,235 | | | | |
| 18 | 171,264 | **177,055** | 173,815 | 175,321 | | | | |
| 20 | 198,706 | 199,460 | 201,769 | **204,278** | | | | |
| 22 | 222,713 | 228,074 | 224,841 | **231,809** | | | | |
| 24 | 246,620 | 248,817 | 248,977 | **253,506** | | | | |
| 26 | 283,971 | 73,900* | 286,239 | **286,847** | | | | |
| 28 | 312,655 | 87,273* | 317,629 | **319,044** | | | | |
| 30 | 343,013 | 94,713* | **352,258** | 344,831 | | | | |
| 32 | 371,143 | 97,059* | 377,531 | **382,508*** | | | | |

**Table 5**
Best results for the IP models on relaxed instances with at least 14 teams ($d_1 + d_2 > 0$). Execution times limited to 3 hour. Best values appear in bold.

| Instance | $n - d_1$ | $\lfloor \frac{n}{2} \rfloor - d_2$ | Best lower bound after 3 hour | | | | Best solution after 3 hour | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\mathcal{F}_1$-M | $\mathcal{F}_1$-BB | $\mathcal{F}_{2+}$-M | $\mathcal{F}_{2+}$-BB | $\mathcal{F}_1$-M | $\mathcal{F}_1$-BB | $\mathcal{F}_{2+}$-M | $\mathcal{F}_{2+}$-BB |
| 14 | 6 | 3 | 149,422 | **150,041** | 149,883 | 149,904 | 167,931 | 167,800 | **164,169** | 165,975 |
| 14 | 5 | 3 | 149,446 | **150,270** | 149,541 | 150,194 | 160,069 | 160,591 | 161,905 | **159,511** |
| 14A | 6 | 3 | 143,138 | 143,207 | **143,931** | 142,456 | 163,872 | 163,622 | **159,054** | 161,201 |
| 14A | 5 | 3 | 142,230 | 142,321 | **143,504** | 142,600 | 158,604 | 158,878 | **154,840** | 159,703 |
| 14B | 6 | 3 | 141,839 | 142,196 | 142,608 | **143,378** | 165,052 | 164,691 | **158,050** | 164,089 |
| 14B | 5 | 3 | 142,117 | 142,237 | **143,706** | 143,147 | 157,907 | 157,174 | **154,739** | 157,724 |
| 14C | 6 | 3 | 140,967 | 141,213 | **141,791** | 140,393 | 163,934 | 159,388 | **153,841** | 157,034 |
| 14C | 5 | 3 | 141,717 | 141,299 | 141,558 | **141,801** | 154,944 | 156,541 | **152,046** | 153,178 |
| 16 | 8 | 2 | 142,005 | 143,420 | 142,019 | **143,840** | 185,505 | **183,167** | 206,299 | 186,108 |
| 16 | 7 | 3 | 143,780 | 144,265 | 144,325 | **145,987** | 207,226 | **188,486** | 190,281 | 195,551 |
| 16 | 7 | 2 | 138,027 | 137,443 | 139,888 | **141,440** | 177,925 | 167,697 | 170,288 | **163,193** |
| 16A | 8 | 2 | 155,542 | 157,013 | 155,743 | **157,972** | 216,598 | 199,684 | 194,050 | **190,233** |
| 16A | 7 | 3 | 156,544 | 159,146 | 157,393 | **160,314** | 210,267 | **205,345** | 209,604 | 215,136 |
| 16A | 7 | 2 | 152,063 | 154,563 | 154,023 | **155,342** | 186,439 | 180,372 | **176,793** | 182,317 |
| 16B | 8 | 2 | 152,675 | 157,356 | 153,734 | **158,035** | **206,044** | 212,300 | 240,845 | 208,859 |
| 16B | 7 | 3 | 154,667 | 157,955 | 155,988 | **158,244** | **215,692** | 237,571 | | 217,261 |
| 16B | 7 | 2 | 151,406 | 155,173 | 152,218 | **155,403** | 203,491 | 192,062 | 191,213 | **184,991** |
| 16C | 8 | 2 | 157,239 | 158,164 | 158,051 | **160,596** | 205,626 | **192,741** | 213,953 | 205,095 |
| 16C | 7 | 3 | 159,916 | 160,438 | 159,970 | **161,838** | 211,880 | **206,505** | 222,039 | 223,262 |
| 16C | 7 | 2 | 156,421 | 156,810 | 157,357 | **158,527** | 186,989 | 192,311 | 182,307 | **182,011** |
| 30 | 5 | 5 | 336,470 | 75,891* | **367,877** | 361,175 | | | | |

**Table 6**
Solutions obtained by the relax-and-fix heuristic on relaxed instances with at least 14 teams. Best values appear in bold.

| Instance | $n-d_1$ | $\lfloor\frac{n}{2}\rfloor-d_2$ | Width of shifting window of binary variables (parameter $b$) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 14 | 7 | 3 | No sol. | 178,226 | 174,652 | 168,408 | 170,629 | 169,981 | 168,408 | **165,573** | 166,942 | 170,250 |
| 14 | 6 | 3 | 165,529 | 165,764 | **159,522** | 163,818 | 159,622 | 162,377 | 160,907 | 159,601 | 164,450 | 163,573 |
| 14 | 5 | 3 | 160,913 | 158,103 | 156,442 | 158,054 | 158,293 | 156,456 | 157,942 | 157,404 | **155,439** | 155,958 |
| 14A | 7 | 3 | 168,422 | 171,938 | 166,357 | 164,136 | 160,830 | 163,471 | 163,860 | 164,784 | **160,046** | 163,433 |
| 14A | 6 | 3 | No sol. | 156,183 | 157,219 | 162,614 | 158,619 | 156,437 | 158,237 | 157,034 | 158,133 | **154,628** |
| 14A | 5 | 3 | 153,955 | 154,251 | 154,218 | 154,891 | 152,588 | 154,441 | **149,331** | 149,956 | 152,855 | 152,490 |
| 14B | 7 | 3 | No sol. | 173,698 | 162,952 | 162,634 | 160,242 | 163,629 | **157,884** | 162,305 | **157,884** | 163,329 |
| 14B | 6 | 3 | 162,891 | 163,161 | 158,818 | 157,927 | 157,222 | **153,611** | 158,291 | 155,358 | 158,382 | 158,735 |
| 14B | 5 | 3 | 156,321 | 155,125 | 153,724 | **150,268** | 150,865 | 151,251 | 150,933 | 150,760 | 150,954 | 153,202 |
| 14C | 7 | 3 | 176,479 | 174,595 | 166,559 | 163,227 | 160,262 | 160,274 | 163,720 | **159,518** | 164,778 | 161,049 |
| 14C | 6 | 3 | 164,223 | 155,523 | 158,304 | **152,158** | 154,954 | 154,026 | 155,025 | 155,435 | 154,366 | 155,200 |
| 14C | 5 | 3 | 153,697 | 151,914 | **149,662** | 150,415 | 149,727 | 150,346 | 151,048 | 150,678 | 150,471 | 151,426 |
| 16 | 8 | 2 | 168,647 | 168,094 | 166,688 | **160,705** | 161,771 | 167,482 | 163,733 | 164,187 | 165,098 | 167,015 |
| 16 | 7 | 3 | No sol. | 181,556 | 177,557 | **169,994** | 174,009 | 170,487 | 172,549 | 179,940 | 173,088 | 177,242 |
| 16 | 7 | 2 | 163,438 | 157,184 | 153,996 | 158,043 | 157,379 | 155,796 | 155,766 | 154,153 | 158,130 | **153,978** |
| 16A | 8 | 2 | 183,344 | 174,841 | 178,320 | 173,956 | 175,092 | 176,821 | **173,950** | 176,664 | 176,266 | 174,546 |
| 16A | 7 | 3 | 186,667 | 193,983 | 185,198 | 188,957 | 183,192 | 192,059 | 182,889 | **181,119** | 188,879 | 188,475 |
| 16A | 7 | 2 | 177,235 | 165,675 | 168,224 | 167,010 | 168,058 | **164,620** | 168,032 | 170,486 | 167,538 | 168,810 |
| 16B | 8 | 2 | 187,840 | 186,326 | 187,514 | 185,223 | 186,853 | 185,541 | **182,673** | 184,750 | 183,238 | 185,253 |
| 16B | 7 | 3 | 207,308 | 194,064 | 189,257 | 191,234 | 188,064 | 191,026 | 187,488 | 188,198 | 189,596 | **187,007** |
| 16B | 7 | 2 | 183,841 | 172,083 | 171,962 | 172,704 | 172,336 | 172,688 | **170,194** | 172,169 | 172,274 | 173,143 |
| 16C | 8 | 2 | 193,358 | 184,873 | 181,205 | **180,221** | 182,568 | 184,103 | 182,706 | 187,598 | 185,487 | 184,753 |
| 16C | 7 | 3 | 189,776 | No sol. | 188,855 | 191,021 | 187,649 | **185,528** | 189,251 | 189,367 | No sol. | 194,055 |
| 16C | 7 | 2 | 174,744 | 174,401 | 172,033 | 171,802 | 170,909 | **169,184** | 170,267 | 171,046 | 172,319 | 170,854 |
| 30 | 5 | 5 | **466,765** | 484,447 | 479,056 | 475,572 | 487,552 | 471,724 | No sol. | No sol. | No sol. | No sol. |

**Table 7**
Execution times (seconds) for the relax-and-fix heuristic on relaxed instances with at least 14 teams.

| Instance | $n-d_1$ | $\lfloor\frac{n}{2}\rfloor-d_2$ | Width of shifting window of binary variables (parameter $b$) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 4, 6, 7 | 1–10 |
| 14 | 7 | 3 | 157 | 254 | 339 | 501 | 716 | 947 | 1099 | 1238 | 1528 | 1815 | 2547 | 8594 |
| 14 | 6 | 3 | 147 | 262 | 302 | 599 | 721 | 953 | 1106 | 1181 | 1424 | 1336 | 2657 | 8031 |
| 14 | 5 | 3 | 125 | 192 | 278 | 493 | 725 | 810 | 989 | 944 | 1053 | 1112 | 2291 | 6721 |
| 14A | 7 | 3 | 179 | 383 | 511 | 722 | 775 | 1132 | 1376 | 1647 | 1703 | 1743 | 3231 | 10,171 |
| 14A | 6 | 3 | 159 | 348 | 485 | 578 | 898 | 1053 | 1356 | 1374 | 1404 | 1546 | 2988 | 9201 |
| 14A | 5 | 3 | 150 | 254 | 397 | 640 | 802 | 931 | 1099 | 1227 | 1316 | 1319 | 2670 | 8134 |
| 14B | 7 | 3 | 170 | 331 | 504 | 497 | 785 | 901 | 1232 | 1405 | 1386 | 1641 | 2631 | 8852 |
| 14B | 6 | 3 | 170 | 303 | 524 | 677 | 925 | 1172 | 1290 | 1453 | 1552 | 1579 | 3139 | 9645 |
| 14B | 5 | 3 | 150 | 272 | 434 | 680 | 917 | 1096 | 1170 | 1170 | 1251 | 1331 | 2946 | 8472 |
| 14C | 7 | 3 | 151 | 334 | 608 | 681 | 889 | 1062 | 1342 | 1346 | 1578 | 1674 | 3085 | 9665 |
| 14C | 6 | 3 | 151 | 271 | 476 | 539 | 736 | 1068 | 1244 | 1371 | 1433 | 1434 | 2851 | 8723 |
| 14C | 5 | 3 | 137 | 227 | 384 | 588 | 775 | 995 | 1045 | 1198 | 1241 | 1223 | 2629 | 7813 |
| 16 | 8 | 2 | 701 | 1398 | 1596 | 1711 | 2022 | 2287 | 2529 | 2968 | 3079 | 3077 | 6528 | 21,369 |
| 16 | 7 | 3 | 722 | 1577 | 1876 | 2231 | 3264 | 4073 | 4417 | 5199 | 5710 | 6073 | 10,721 | 35,143 |
| 16 | 7 | 2 | 771 | 1293 | 1620 | 2169 | 2572 | 2800 | 2901 | 3239 | 3628 | 3584 | 7869 | 24,577 |
| 16A | 8 | 2 | 848 | 1333 | 1676 | 1729 | 2200 | 2259 | 2351 | 2818 | 3045 | 2954 | 6339 | 21,213 |
| 16A | 7 | 3 | 954 | 1690 | 1952 | 2718 | 3311 | 3537 | 4473 | 4830 | 5591 | 6506 | 10,728 | 35,560 |
| 16A | 7 | 2 | 821 | 1110 | 1984 | 2099 | 2272 | 2379 | 3089 | 3191 | 3469 | 3819 | 7567 | 24,232 |
| 16B | 8 | 2 | 927 | 1633 | 1861 | 1985 | 2325 | 2577 | 2934 | 2968 | 3305 | 3396 | 7495 | 23,911 |
| 16B | 7 | 3 | 935 | 1464 | 1939 | 2536 | 2981 | 4265 | 4296 | 4519 | 5534 | 5832 | 11,097 | 34,301 |
| 16B | 7 | 2 | 872 | 1316 | 1950 | 2128 | 2499 | 2948 | 2996 | 3525 | 3438 | 4112 | 8072 | 25,784 |
| 16C | 8 | 2 | 844 | 1326 | 1563 | 1892 | 1842 | 2211 | 2671 | 2791 | 3187 | 3557 | 6774 | 21,884 |
| 16C | 7 | 3 | 764 | 1188 | 1812 | 2433 | 2721 | 3235 | 3991 | 4615 | 1122 | 4836 | 9659 | 26,717 |
| 16C | 7 | 2 | 834 | 1151 | 1558 | 1780 | 1948 | 2437 | 2729 | 2893 | 3012 | 3200 | 6947 | 21,543 |
| 30 | 5 | 5 | 16,352 | 13,817 | 11,825 | 10,821 | 8397 | 14,917 | 5208 | 4841 | 5033 | 6455 | 30,946 | 97,666 |

and gets reasonably close in the remaining cases, except for instance 30. The best lower bounds shown in Table 4 are better than the best known bounds reported in the TUP benchmark, for *all* instance sizes. In particular, the benchmark reports no lower bounds for instances with more than 16 teams.

When it comes to quality of feasible solutions, no formulation finds solutions to hard instances with more than 14 teams (no such solutions are known, to the best of our knowledge). Formulation $\mathcal{F}_{2+}$-M finds three out of the four best 14-team solutions, all of which are better than the best known solutions found by *exact* methods reported in Trick and Yildiz (2012).

### 5.2. IP results on relaxed instances: Lower and upper bounds

Table 5 summarizes the performance of our IP models on relaxed instances ($d_1 + d_2 > 0$) with at least 14 teams. Because calculating $k^*$ for the 30-team instance with $\mathcal{F}_1$-BB takes too long, we report the best lower bound found within 3 hours and mark it with an "$*$". Except for the first two 14-team instances, the lower bounds provided by $\mathcal{F}_{2+}$ models are always better than those provided by $\mathcal{F}_1$ models. The $\mathcal{F}_{2+}$-BB model is responsible for 14 out of the 21 best lower bounds, and its solution is reasonably close to the best bound in the remaining cases. Formulation $\mathcal{F}_{2+}$-M does well

**Table 8**
Comparison between the best results obtained by our exact and heuristic approaches and the best results from literature on hard and relaxed instances with at least 14 teams.

| Instance | $n-d_1$ | $\lfloor \frac{n}{2} \rfloor - d_2$ | Lower bounds | | | | Solution values | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Lit. Best | $\mathcal{F}_{2+}^R$-BB | $\mathcal{F}_{2+}$-BB | $\mathcal{F}$ Best | Lit. Best | $\mathcal{F}$ Best | RF Worst | RF 4,6,7 | RF Best |
| 14 | 7 | 3 | 141,253 | 143,089 | 150,871 | 150,871 | 166,964 | 174,715 | 178,226 | 168,408 | 165,573 |
| 14 | 6 | 3 | 141,064 | 142,716 | 149,904 | 150,041 | 173,681 | 164,169 | 165,764 | 160,907 | 159,522 |
| 14 | 5 | 3 | 141,134 | 142,478 | 150,194 | 150,270 | 165,558 | 159,511 | 160,913 | 156,456 | 155,439 |
| 14A | 7 | 3 | 133,279 | 135,149 | 141,308 | 143,517 | 160,407 | 165,968 | 171,938 | 163,471 | 160,046 |
| 14A | 6 | 3 | 133,194 | 134,971 | 142,456 | 143,931 | 164,857 | 159,054 | 162,614 | 156,437 | 154,628 |
| 14A | 5 | 3 | 133,023 | 134,884 | 142,600 | 143,504 | 162,380 | 154,840 | 154,891 | 149,331 | 149,331 |
| 14B | 7 | 3 | 131,373 | 131,684 | 142,614 | 142,614 | 161,129 | 168,659 | 173,698 | 157,884 | 157,884 |
| 14B | 6 | 3 | 130,799 | 131,542 | 143,378 | 143,378 | 168,476 | 158,050 | 163,161 | 153,611 | 153,611 |
| 14B | 5 | 3 | 130,628 | 131,301 | 143,147 | 143,706 | 160,443 | 154,739 | 156,321 | 150,268 | 150,268 |
| 14C | 7 | 3 | 126,843 | 131,163 | 138,601 | 141,268 | 159,461 | 164,512 | 176,479 | 160,274 | 159,518 |
| 14C | 6 | 3 | 126,613 | 130,921 | 140,393 | 141,791 | 166,395 | 153,841 | 164,223 | 152,158 | 152,158 |
| 14C | 5 | 3 | 126,427 | 130,556 | 141,801 | 141,801 | 163,662 | 152,046 | 153,697 | 150,346 | 149,662 |
| 16 | 8 | 4 | 134,471 | 134,151 | 150,523 | 151,748 | | | | | |
| 16 | 8 | 2 | 134,347 | 132,563 | 143,840 | 143,840 | 178,775 | 183,167 | 168,647 | 160,705 | 160,705 |
| 16 | 7 | 3 | 121,933 | 127,593 | 145,987 | 145,987 | 185,966 | 188,486 | 181,556 | 169,994 | 169,994 |
| 16 | 7 | 2 | 121,670 | 126,946 | 141,440 | 141,440 | 166,114 | 163,193 | 163,438 | 155,766 | 153,978 |
| 16A | 8 | 4 | 148,377 | 147,551 | 166,101 | 166,626 | | | | | |
| 16A | 8 | 2 | 146,992 | 145,595 | 157,972 | 157,972 | 188,432 | 190,233 | 183,344 | 173,950 | 173,950 |
| 16A | 7 | 3 | 137,178 | 142,945 | 160,314 | 160,314 | 199,016 | 205,345 | 193,983 | 182,889 | 181,119 |
| 16A | 7 | 2 | 137,806 | 142,056 | 155,342 | 155,342 | 172,728 | 176,793 | 177,235 | 164,620 | 164,620 |
| 16B | 8 | 4 | 146,646 | 146,805 | 161,882 | 162,251 | | | | | |
| 16B | 8 | 2 | 145,058 | 145,383 | 158,035 | 158,035 | 201,039 | 206,044 | 187,840 | 182,673 | 182,673 |
| 16B | 7 | 3 | 139,833 | 141,838 | 158,244 | 158,244 | 202,395 | 215,692 | 207,308 | 187,488* | 187,007 |
| 16B | 7 | 2 | 139,742 | 141,552 | 155,403 | 155,403 | 184,923 | 184,991 | 183,841 | 170,194 | 170,194 |
| 16C | 8 | 4 | 145,012 | 152,698 | 164,235 | 165,431 | | | | | |
| 16C | 8 | 2 | 144,398 | 150,451 | 160,596 | 160,596 | 202,023 | 192,741 | 193,358 | 180,221 | 180,221 |
| 16C | 7 | 3 | 142,467 | 148,932 | 161,838 | 161,838 | 213,157 | 206,505 | 194,055 | 185,528 | 185,528 |
| 16C | 7 | 2 | 142,399 | 148,481 | 158,527 | 158,527 | 181,013 | 182,011 | 174,744 | 169,184 | 169,184 |
| 30 | 5 | 5 | | 336,124 | 361,175 | 367,877 | 483,224 | | 487,552 | 471,724* | 466,765 |

with 14-team instances, finding 4 out of the 8 best lower bounds, and it also finds the best bound for the 30-team instance. When it comes to feasible solutions, $\mathcal{F}_{2+}$-M manages to find 8 out of the 21 best solutions.

The best lower bounds shown in Table 5 are better than the best known bounds reported in the TUP benchmark, for *all* instance sizes. In addition, 18 out of the 20 best solutions shown in Table 5 are better than the best solutions found by *exact* methods reported in Trick and Yildiz (2012). More importantly, 11 of these solutions (underlined numbers) are better than the best known solution found by *any* method, according to the TUP benchmark. In Section 5.3, we show that our heuristic can do even better than that.

### 5.3. Heuristic results

We use CPLEX to solve the optimization models inside the relax-and-fix heuristic (see Table A.9 in A for the parameter settings we used). We apply the relax-and-fix heuristic with $b = 1, 2, \ldots, 10$ to the instances used in our previous experiments. The heuristic finds an optimal solution to all but one of the hard instances with at most 10 teams. (The best solution obtained by the heuristic for instance 10C is 43,193, which is close to the optimal distance of 43,149.) The total time spent to obtain these solutions for all values of $b$ is at most 1 second for instances with 4 and 6 teams, at most 10 seconds for instances with 8 teams, at most 2 minutes for instances 10, 10A, and 10B, and around 3 minutes for instance 10C. Tables 6 and 7 present, respectively, the solution values and execution times for hard and relaxed instances with at least 14 teams. Unfortunately, similarly to previous attempts reported in the literature, the relax-and-fix heuristic does not manage to find feasible solutions to hard instances with more than 14 teams. For the instances to which it manages to find a feasible solution, the heuristic appears to be reliable, obtaining solutions in 240 out of 250 runs (25 instances, 10 values of $b$). Most of the best solutions (14 out of 25) are found with $b = 4, 6$, or 7. Finally, we also include in Table 7 the total cumulative time to find the solutions for $b \in \{4, 6, 7\}$ (column "4,6,7") and for $b \in \{1, 2, \ldots, 10\}$ (column "1–10").

### 5.4. Best results

We now compare the best results obtained by our approaches (exact and heuristic) with the best results from the literature, which are published in Trick and Yildiz (2011); Trick and Yildiz (2012); Trick et al. (2012); and Trick (2013). The most relevant comparison for hard instances with no more than 10 teams are the times required to reach optimality. Table 3 indicates that our strengthened formulation $\mathcal{F}_{2+}$ does better than $\mathcal{F}_1$ on these small instances, and that a good choice of $k$ (e.g. middle-round) pays off.

The main improvements we achieve are on instances with at least 14 teams, which have no known optimal solutions. Table 8 summarizes the best results for these instances (an "$*$" indicates that the time limit was exceeded; see Table 7). The best lower bounds and solutions from the literature were obtained within 3 hours of computation for instances with 14 and 16 teams, and within 5 hours for the relaxed 30-team instance (solution only; no lower bound is reported). The lower bounds provided by $\mathcal{F}_{2+}^R$-BB are already better than the best ones in the literature on 25 out of 29 instances. They can be calculated in less than 1 minute for instances with at most 16 teams, and in approximately 2 hours for the instance with 30 teams. Within 3 hours of computation, the best lower bounds found during branch-and-cut with $\mathcal{F}_{2+}$-BB beat all known lower bounds by margins ranging between 8 and 24 thousand miles. The "$\mathcal{F}$ Best" column under "Lower bounds" shows the best bounds we managed to find over the entire range of models we considered. These bounds are better than the $\mathcal{F}_{2+}$-BB bounds on 13 out of 29 instances.

On the right half of Table 8 we compare the quality of the best known solutions in the literature with those obtained by our IP models and by the relax-and-fix heuristic. All of our best solutions are obtained by some configuration of the relax-and-fix heuristic (column "RF Best"). The only instance whose solution we do not manage to improve is 14C-7-3 (by a mere 57 miles). For the remaining 24 instances with known feasible solutions, our best solutions are between 0.36 and 27.6 thousand miles shorter. When compared with the best solutions obtained by the IP models (column "$\mathcal{F}$ Best"), the best relax-and-fix solutions are between 1.68 and 28.6 thousand miles shorter. However, except for instances with 14 teams, this comparison is unfair because finding the best relax-and-fix solution over all values of $b \in \{1, 2, \ldots, 10\}$ takes much more time than the limits of 3 and 5 hours (see Table 7). Therefore, we include column "RF 4,6,7" which has the best solution obtained over three runs of relax-and-fix with $b = 4, 6$, and 7. These three runs obtain most of our best results (14 out of 25) and, in total, stay within the time limits with two exceptions: instance 16B-7-3 (3-hour limit exceeded by 5 minutes), and instance 30. When compared to the best known solutions in the literature, solutions under "RF 4,6,7" have less mileage in 22 out of 25 cases. The worst solutions obtained by the relax-and-fix heuristic (column "RF Worst") are better than the best known solutions in 18 out of 25 cases. Finally, for the relaxed instance with 30 teams, 4 out of the 6 solutions found by the relax-and-fix heuristic are better than the best one in the literature, with the greatest improvement being equal to 16,459 miles.

## 6. Conclusions and future work

By strengthening an existing IP formulation for the traveling umpire problem (TUP), we obtain an optimization model that not only solves more quickly than its original version, but also provides better lower and upper bounds for instances in the TUP benchmark. This new formulation plays a crucial role in our implementation of a relax-and-fix heuristic for the problem, because each iteration of the heuristic cannot dedicate too much time to solving its intermediate IP models. As a result, we improve all known lower bounds for instances in the benchmark, as well as 24 of the 25 best known upper bounds. Moreover, we are the first to provide strong lower bounds for instances with more than 16 teams.

The TUP remains a very difficult problem, with many small instances lacking feasible solutions. We believe that a combination of exact and heuristic methods is a promising research direction. The TUP formulations deserve a deeper polyhedral study, and our relax-and-fix heuristic can be modified in several ways. For example, the sliding windows of binary variables can take different shapes, focus first on problematic areas of the schedule (a bottleneck approach), and/or use randomization. Finally, not much attention has been given to improving the CP model of Trick and Yildiz (2011), and a more effective version of that model (e.g. with fancier search routines) could become part of a relax-and-fix heuristic as well.

We hope that the improvements in solution quality presented in this paper will spark the interest of other researchers in tackling this challenging problem.

**Table A.9**
CPLEX branch-and-cut parameters used to solve the models inside the relax-and-fix heuristic.

| Parameter | ⩽18 Teams | >18 Teams |
|---|---|---|
| Number of parallel threads | 1 | 1 |
| MIP node limit | 100 | 100 |
| MIP repeat presolve | Off | Off |
| MIP heuristic frequency | 2 | |
| RINS heuristic frequency | 20 | |
| Feasibility pump heuristic | On | |
| SubMIP node limit | 10 | |
| MIP probing level | Very aggressive | |
| MIP dive strategy | Probing dive | |
| MIP integer solution limit | | 1 |

## Appendix A

See Table A.9.

## References


Ernst, A., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D. (2004). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research, 127*, 21–144.

Evans, J. R. (1988). A microcomputer-based decision support system for scheduling umpires in the american baseball league. *Interfaces, 18*, 42–51.

Evans, J. R., Hebert, J. E., & Deckro, R. F. (1984). Play ball – The scheduling of sports officials. *Perspectives in Computing, 4*, 18–29.

Farmer, A., Smith, J. S., & Miller, L. T. (2007). Scheduling umpire crews for professional tennis tournaments. *Interfaces, 37*, 187–196.

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

Trick, M. A. (2013). Traveling umpire problem: Data sets and results. <http://mat.tepper.cmu.edu/TUP> (Last accessed: 07.08.13).

Trick, M. A., & Yildiz, H. (2007). Benders' cuts guided large neighborhood search for the traveling umpire problem. In P. Van Hentenryck & L. Wolsey (Eds.), *Proceedings of the fourth conference on integration of AI and OR techniques in constraint programming for combinatorial optimization problems (CP–AI–OR). Lecture notes in computer science* (Vol. 4510, pp. 332–345). Springer-Verlag.

Trick, M. A., & Yildiz, H. (2011). Benders' cuts guided large neighborhood search for the traveling umpire problem. *Naval Research Logistics, 58*, 771–781.

Trick, M. A., & Yildiz, H. (2012). Locally optimized crossover for the traveling umpire problem. *European Journal of Operational Research, 216*, 286–292.

Trick, M. A., Yildiz, H., & Yunes, T. (2012). Scheduling major league baseball umpires and the traveling umpire problem. *Interfaces, 42*, 232–244.

Wauters, T. (2013a). Een nieuwe (meta)heuristiek voor het Traveling Umpire Problem. <www.kuleuven-kulak.be/nl/onderzoek/Wetenschappen/Informatica/travelingumpireproblem> Katholieke Universiteit Leuven, Belgium. Bachelor's Thesis Proposal (abstract in Dutch; see <www.kuleuven-kulak.be/nl/onderzoek/Wetenschappen/Informatica/bachelorthesis>).

Wauters, T., (2013b). Solution posted on the traveling umpire problem benchmark web site as of August 2013.

Wolsey, L. A. (1998). *Integer programming*. Wiley-Interscience.

Wright, M. B. (1991). Scheduling english cricket umpires. *Journal of the Operational Research Society, 42*, 447–452.

Yavuz, M., İnan, U. H., & Fığlalı, A. (2008). Fair referee assignments for professional football leagues. *Computers & Operations Research, 35*, 2937–2951.

Yildiz, H. (2008). Methodologies and applications for scheduling, routing & related problems. Ph.D. thesis Tepper School of Business. Carnegie Mellon University.