A Benders Approach for Computing Improved Lower Bounds for the Mirrored Traveling Tournament Problem

Kevin K. H. Cheung  
School of Mathematics and Statistics  
Carleton University  
1125 Colonel By Drive  
Ottawa, ON K1S 5B6  
Canada  
E-mail: kcheung@math.carleton.ca  
url: \url{http://www.math.carleton.ca/~kcheung}

April 3, 2008

Abstract

A Benders approach for computing lower bounds for the mirrored Traveling Tournament Problem is proposed. The method obtained improved lower bounds for a number of benchmark instances at the Challenge Traveling Tournament Problems homepage \url{http://mat.gsia.cmu.edu/TOURN/}.

Keywords: Timetabling, Sports Scheduling, Integer programming, Logic-based Benders cuts

1 Introduction

The Traveling Tournament Problem (TTP) is a class of sports scheduling proposed by Easton et al. [8]. Benchmark instances for the TTP are hosted at the Challenge Traveling Tournament Problems homepage \url{http://mat.gsia.cmu.edu/TOURN/}. Over the years, various heuristics and solution methods have been proposed and good schedules to these benchmark instances have been obtained (see for example [1, 2, 9, 10, 14, 15, 18, 19] and the survey [16]). Schedules with matching lower bounds have been found for many of the Constant Distance instances with and without the mirror requirement (see [10, 18, 20]) but for only a handful of the more general instances. One general lower bound that can be computed within a reasonable amount of time even for a large number of teams is the Independent Lower Bound (ILB) introduced by Easton et al. [8]. By taking into account the total number of trips that all the teams must make, Urrutia et al. [21] improved the ILB to give what they call the Minimum Number of Trips Lower
Bound (MNTLB). Inspired by the success of a Benders approach in the work of Rasmuseen and Trick [18] and the work of Codato and Fischetti [6], we propose a method for improving the MNTLB using logic-based Benders decomposition and report on the computational results.

2 The Mirrored Traveling Tournament Problem

Given \( n \) teams where \( n \) is even, a double round-robin (DRR) tournament is a set of games in which every team plays every other team exactly once at home and once away in \( 2(n-1) \) time slots such that each team plays exactly one game in each slot. Distances between team venues are given. Each team begins at its home venue and travels to play its games at the appropriate venues specified by the schedule and returns (if necessary) to the home venue at the end of the schedule. The TTP is the problem of finding a schedule for a DRR tournament on the set of teams, let \( S \) be the set of teams, let \( A \) denote the distance between the venue of team \( i \) and the venue of team \( j \). For each \( t \in T \), define \( h_t^1 \) and \( h_t^2(n-1)+1 \) to be the constant 1. With decision variables \( h_t^i \in \{0,1\}^S \) for each \( t \in T \), \( x^s \in \{0,1\}^A \) for each \( s \in S \), and \( u^t \in \mathbb{R}^E \) for each \( t \in T \), the mTTP can be formulated as follows:

\[
\text{min } \sum_{t \in T} \sum_{e \in E} c_{e} u_{e}^t \quad \text{(MIP)}
\]

\[
\text{s.t. } \begin{align*}
 u_{t(i,j)}^1 &\geq x_{t(i,i)}^s + x_{t(i,j)}^{s+1} - 1 & \forall t \in T, \forall s \in S \setminus \{2(n-1)\}, \\
 & \quad \forall i, j \in T \setminus \{t\} \text{ with } i \neq j, \\
 u_{t(i,i)}^1 &\geq h_{t-1}^i + 2(x_{t(i,i)}^{s(t)} - 1) + h_{t+1}^i \forall t, i \in T \text{ with } i \neq t, \forall s \in S, \\
 \sum_{i \in T \setminus \{t\}} x_{t(i,t)}^s + x_{t(i,s)}^s &\geq 1 & \forall s \in S, \forall t \in T, \\
 \sum_{s \in S} x_{e}^s &\geq 1 & \forall e \in A, \\
x_{t(i,j)}^s &\geq x_{t(j,i)}^{s+n-1} & \forall (i, j) \in A, \forall s = 1, \ldots, n-1, \\
 1 &\leq \sum_{j=0}^{3} h_{t+j}^s & \forall t \in T, s = 1, \ldots, 2(n-1) - 3, \\
h_{t}^s &\geq \sum_{i \in T \setminus \{t\}} x_{t(i)}^s & \forall s \in S, \forall t \in T \\
x^s &\in \{0,1\}^A & \forall s \in S, \\
h^t &\in \{0,1\}^S & \forall t \in T, \\
u^t &\geq 0 & \forall t \in T. 
\end{align*}
\]
Remark. For the general TTP problem, one simply replaces (6) with the following constraints for the no-repeater requirement:

$$\sum_{m=0}^{1} x_{i,j}^{t+m} + x_{i',j}^{t+m} \leq 1 \quad \forall (i,j) \in A, \forall s \in S \setminus \{2(n-1)\}.$$ 

The interpretation of the variables is as follows: $h_t^i = 1$ if and only if team $t$ plays at home in slot $s$, and $x_{i,j}^s = 1$ if and only if, in slot $s$, team $i$ plays an away game against team $j$ at the venue of team $j$. Constraints (4) ensure that each team plays exactly one game in each slot. Constraints (5) ensure that each team plays an away game against each of the other teams in exactly one slot. Constraints (6) enforce the mirror requirement. Constraints (7) ensure that no more than three consecutive games are all home games or all away games. Since (MIP) is a minimization problem and $c \geq 0$, it is not difficult to see that there is an optimal solution $\tilde{u}, \tilde{h}, \tilde{x}$ such that for every $t \in T$, $\tilde{u}_{i,j}^t \in \{0,1\}$ for all $i,j \in T \setminus \{t\}$ with $i \neq j$, and $\tilde{u}_{i,t,1}^t \in \{0,1,2\}$ for all $i \in T \setminus \{t\}$. In particular, $\tilde{u}_{i,j}^t = 1$ if and only if there exists $s$ such that either $\tilde{x}_{i,s}^t = x_{i,j}^{t+1} = 1$ or $\tilde{x}_{i,s}^{t+1} = \tilde{x}_{i,j}^s = 1$; that is, team $t$ plays away against team $i$ and team $j$ in two consecutive slots. In addition, $\tilde{u}_{i,t,1}^t = 2$ if and only if there exists a slot in which team $t$ plays away against team $i$ and team $t$ is at home in the preceding and the succeeding slots; $\tilde{u}_{i,t,1}^t = 1$ if and only if there exists a slot in which team $t$ plays away against team $i$ and team $t$ is at home in either the preceding slot or the succeeding slot (but not both). Hence, if one adds up the entries in $\tilde{u}^t$ for a particular team $t$, one gets the total number of trips team $t$ has to make. For $W \subseteq T$, define $\delta(W) := \{i,j \in E : i \in W \text{ and } j \notin W\}$. We abbreviate $\delta(\{i\})$ as $\delta(i)$. As the upper bound on the number of consecutive away games is three, for each $t \in T$, $\tilde{u}^t$ satisfies:

$$\sum_{e \in \delta(i)} \tilde{u}^t_e = 2 \quad \forall i \in T \setminus \{t\},$$

$$\sum_{e \in \delta(W)} \tilde{u}^t_e \geq 2|W|/3 \quad \forall W \subseteq T \setminus \{t\},$$

$$\tilde{u}^t_e \in \{0,1\} \quad \forall e \in E \setminus \delta(t),$$

$$\tilde{u}^t_e \in \{0,1,2\} \quad \forall e \in \delta(t).$$

From these constraints, one can see that $\tilde{u}^t$ corresponds to a route in a capacitated vehicle routing problem (CVRP) with $t$ as depot, $T \setminus \{t\}$ as the set customers each with unit demand, and vehicle capacity 3. Using this observation, Eastman et al. [9] introduced the Independent Lower Bound (ILB) for the TTP given by the optimal value of

$$\text{(ILBIP)} \quad \min \sum_{t \in T} \sum_{e \in E} c_e u_e^t$$

s.t. \begin{align*}
\sum_{e \in \delta(i)} u_e^t &= 2 \quad \forall t \in T, \forall i \in T \setminus \{t\}, \\
\sum_{e \in \delta(W)} u_e^t &\geq 2|W|/3 \quad \forall t \in T, \forall W \subseteq T \setminus \{t\}, \\
\forall e \in E \setminus \delta(t), \quad u_e^t &\in \{0,1\}, \\
\forall e \in \delta(t), \quad u_e^t &\in \{0,1,2\}.
\end{align*}
Urrutia et al. [21] proposed an improved lower bound, called the Minimum Number of Trips Lower Bound (MNTLB), by adding the constraint
\[
\sum_{t \in T} \sum_{e \in E} u^t_e \geq M_n
\]
to \((ILBIP)\) where \(M_n\) denotes the minimum number of trips in any valid schedule for the mTTP with \(2n\) teams. (They also defined the MNTLB for the general TTP similarly.) The value for \(M_n\) can be the optimal value or the best known lower bound for the Constant Distance instance on \(2n\) teams. We point out that the calculation of the MNTLB in Urrutia et al. [21] is done in two stages as opposed to solving a single mixed-integer linear programming program. Their approach has the advantage of requiring less memory as the computations can be decomposed. However, the formulation above makes it easy for the addition of Benders cuts.

3 Methodology

We propose a method for obtaining improved lower bounds for some benchmark instances of the mTTP using logic-based Benders decomposition. A special case of logic-based Benders decomposition was first introduced by Hooker and Yen [13] in the context of logic circuit verification. The approach was developed into generality by Hookers and Ottoson [12], making it possible to apply Benders decomposition, for example, on mixed-integer linear programming problems not having a linear programming slave problem. A synthesis of various ideas on logic-based Benders decomposition and inference duality is given in the book by Hooker [11]. In this section, we describe the approach specialized to our problem. We point out that even though the method described below should work in theory for the general TTP, we were able to obtain improvements only for mTTP instances. Hence, we will focus on the mTTP from now on.

As an initial attempt, we took the problem for computing the MNTLP described in the previous section as the initial master problem:

\[
\begin{align*}
\min & \quad \sum_{t \in T} \sum_{e \in E} c_e u^t_e \\
\text{s.t.} & \quad \sum_{t \in T} \sum_{e \in E} u^t_e \geq M_n \\
& \quad \sum_{e \in \delta(i)} u^t_e = 2 \quad \forall t \in T, \forall i \in T \setminus \{t\}, \\
& \quad \sum_{e \in \delta(W)} u^t_e \geq 2|W|/3 \quad \forall t \in T, \forall W \subseteq T \setminus \{t\}, \\
& \quad u^t_e \in \{0, 1\} \quad \forall t \in T, \forall e \in E \setminus \delta(t), \\
& \quad u^t_e \in \{0, 1, 2\} \quad \forall t \in T, \forall e \in \delta(t).
\end{align*}
\]

The slave problem is the problem of determining if there exist \(x\) and \(h\) satisfying (2)—(10) for a solution \(u\) of the master problem. If a solution exists, then we have found an optimal solution to \((MIP)\). Otherwise, a Benders cut is added to the master problem that “cuts off” the current solution \(u\). This scheme can be implemented as described in [12]. Unfortunately, the slave
problem as described above turned out to be difficult for CPLEX to solve even for \( n = 10 \). In addition, that not all entries of \( u \) are necessarily binary was a bit inconvenient to work with.

We then turned to a natural extended formulation of the master problem and were able to obtain positive results. For each team, instead of variables indexed by unordered pairs of teams (the variables \( u \)) to keep track of the trips to be made, we have variables indexed by all the possible subroutes for the underlying CVRP. (There are \( O(n^3) \) of these variables since the length of a subroute is at most three as no more than three consecutive away games are allowed.) In particular, for each \( t \in T \), we have the binary variables \( y_{it}^t \) for all \( i \in T \setminus \{ t \} \), \( y_{i_1i_2}^t \) for all \( i_1, i_2 \in T \setminus \{ t \} \) with \( i_1 < i_2 \), and \( y_{i_1i_2i_3}^t \) for all distinct \( i_1, i_2, i_3 \in T \setminus \{ t \} \) with \( i_1 < i_3 \). The interpretation of these variables is as follows: \( y_{it}^t = 1 \) if and only if the subroute that leaves the depot \( t \) and visits customer \( i \) then returns to the depot \( t \) is selected; \( y_{i_1i_2}^t = 1 \) if and only if the subroute that leaves \( t \) and visits \( i_1 \) then \( i_2 \), or \( i_2 \) then \( i_1 \), and then returns to \( t \) is selected; \( y_{i_1i_2i_3}^t = 1 \) if and only if the subroute that leaves \( t \) and visits \( i_1, i_2 \), and then \( i_3 \), or \( i_3, i_2 \), and then \( i_1 \), and then returns to \( t \) is selected.

For each \( t \in T \) and \( e \in E \), define

\[
U_e^t := \begin{cases} 
2y_{it}^t + \sum_{i_1, i_2 \in T \setminus \{ t \}} y_{i_1i_2}^t + \sum_{i_1, i_2, i_3 \in T \setminus \{ t \}} y_{i_1i_2i_3}^t & \text{if } e = \{ t, i \} \text{ for some } i \in T \setminus \{ t \}, \\
y_{ij}^t + \sum_{i_1, i_2, i_3 \in T \setminus \{ t \}} y_{i_1i_2i_3}^t & \text{if } e = \{ i, j \} \text{ for some } i, j \in T \setminus \{ t \}, i < j.
\end{cases}
\]

Then the initial master problem can be formulated as:

\[
\begin{align*}
\min \sum_{t \in T} \sum_{e \in E} c_{e} U_e^t \\
\text{s.t.} \sum_{t \in T} \sum_{e \in E} U_e^t & \geq M_n \\
y_{it}^t + \sum_{i_1, i_2 \in T \setminus \{ t \}} y_{i_1i_2}^t + \sum_{i_1, i_2, i_3 \in T \setminus \{ t \}} y_{i_1i_2i_3}^t & = 1 \quad \forall t \in T, \forall i \in T \setminus \{ t \}, \\
\sum_{t \in T \setminus \{ t \}} y_{it}^t + \sum_{i_1, i_2 \in T \setminus \{ t \}} y_{i_1i_2}^t + \sum_{i_1, i_2, i_3 \in T \setminus \{ t \}} y_{i_1i_2i_3}^t & \geq \left( \frac{n-1}{3} \right) \quad \forall t \in T, \\
y_{it}^t & \in \{0, 1\} \quad \forall t \in T, \forall i \in T \setminus \{ t \}, \\
y_{i_1i_2}^t & \in \{0, 1\} \quad \forall t \in T, \forall i_1, i_2 \in T \setminus \{ t \}, i_1 < i_2, \\
y_{i_1i_2i_3}^t & \in \{0, 1\} \quad \forall t \in T, \forall \text{ distinct } i_1, i_2, i_3 \in T \setminus \{ t \}, i_1 < i_3.
\end{align*}
\]

Note that the number of constraints in the above formulation is polynomial in \( n \). In our computational experiments, we could solve \((MP)\) using CPLEX for the NFL30 instance in just a few minutes with aggressive cut generation. To motivate the formulation of the slave problem, we first consider an example.
Example 3.1. The cost matrix for the instance NL4 is

\[
\begin{bmatrix}
0 & 745 & 665 & 929 \\
745 & 0 & 80 & 337 \\
665 & 80 & 0 & 380 \\
929 & 337 & 380 & 0 \\
\end{bmatrix}
\]

An optimal solution to (MP) found by CPLEX is

\[
y_{3,2,4}^1 = y_{3,1,4}^2 = y_{1}^3 = y_{2,4}^3 = y_{1,3,2}^4 = 1
\]

with the rest of the variables equal to zero. A schedule that realizes \( y \) needs

- team 1 to play three consecutive away games against teams 3, 2, and 4 in that order or in reverse order;
- team 2 to play three consecutive away games against teams 3, 1, and 4 in that order or in reverse order;
- team 3 to leave from home to play an away game against team 1 and then returns to home at some point in the schedule, and to play two consecutive away games against teams 2 and 4 in that order or in reverse order at some other point in the schedule;
- team 4 to play three consecutive away games against teams 1, 3, and 2 in that order or in reverse order.

Suppose that we have a schedule that satisfy the above conditions. Since reversing the slots of the schedule of an mTTP still gives a schedule of the same mTTP with the same cost, we may assume that team 1 plays away against teams 3, 2, and 4 in slots \( s, s+1, \) and \( s+2 \), respectively, for some \( s \in \{1, 2, 3, 4\} \). As team 4 plays at home against 1 in slot \( s+2 \), team 4 plays away against team 1 in slot \( s \) or slot \( s+1 \). If it is the former, then team 4 must also play away against team 3 in slot \( s \), which is impossible since team 1 plays away against team 3 in the same slot. If it is the latter, then \( s = 1 \) and team 4 must play at home against team 2 in slot 1, implying that team 2 plays away against team 1 in slot 2, contradicting that team 1 plays away against team 2 in the same slot. Hence, there is no schedule realizing \( y \) if \( y_{1,2,4}^1, y_{3,1,4}^2, y_{1,3,2}^4 \) are all equal to 1. Therefore, the inequality \( y_{1,2,4}^1 + y_{3,1,4}^2 + y_{1,3,2}^4 \leq 2 \) can be added to the master problem.

We now describe the slave problem given a solution \( y \) to the master problem. For each \( t \in T \), the slave problem has the following binary variables in addition to the variables \( x \) and \( h \) in (MIP):

- \( z_{s,i,j}^t \) for all \( s \in S \) and \( i \in S_1^t(y) := \{j : y_{ji}^t = 1\} \);
- \( z_{s,ij,1}^t \) and \( z_{s,ij,2}^t \) for all \( s \in S \setminus \{2(n-1)\} \) and \( i_1, i_2 \in T \setminus \{t\} \) such that \( i_1 i_2 \in S_2^t(y) := \{j_1 j_2 : y_{j_1j_2}^t = 1\} \);
- \( z_{s,ij,1}^t \) and \( z_{s,ij,2}^t \) for all \( s \in S \setminus \{2(n-1) - 1, 2(n-1)\} \) and \( i_1, i_2, i_3 \in T \setminus \{t\} \) such that \( i_1 i_2 i_3 \in S_3^t(y) := \{j_1 j_2 j_3 : y_{j_1j_2j_3}^t = 1\} \).

For convenience, we define \( z_{s,ij}^t := 0 \) for \( s \not\in S \setminus \{2(n-1)\} \), and \( z_{s,ij,1}^t := 0 \) for \( s \not\in S \setminus \{2(n-1) - 1, 2(n-1)\} \). The interpretation of the variables is as follows: \( z_{s,i}^t = 1 \) if and only if team \( t \) is at home in slot \( s \) and slot \( s+1 \) and plays away against team \( i \) in slot \( s \). \( z_{s,ij}^t = 1 \) if and
away against team \( t \) in slot \( s \) and away against team \( j \) in slot \( s+1 \). \( z_{s,jk}^t = 1 \) if and only if team \( t \) is at home in slot \( s-1 \) and slot \( s+3 \) and plays away against team \( i \) in slot \( s \), away against team \( j \) in slot \( s+1 \), and away against team \( k \) in slot \( s+2 \).

For each \( t \in T \),

- let \( C_y^t(i) \), where \( i \in S^t_1(y) \), denote the set of constraints:
  \[
  \sum_{s=1}^{2(n-1)} z_{s,i}^t = 1,
  \]
  \[
  x_{s,t}^t = z_{s,i}^t \quad \forall \ s \in S,
  \]
  \[
  z_{s,i}^t \in \{0,1\} \quad \forall \ s \in S;
  \]

- let \( C_y^t(i_1i_2) \), where \( i_1i_2 \in S^t_2(y) \), denote the set of constraints:
  \[
  \sum_{s=1}^{2(n-1)-1} (z_{s,1,i_1i_2}^t + z_{s,1,i_2i_1}^t) = 1,
  \]
  \[
  x_{s,t}^t = s_{s,1,i_1i_2}^t + z_{s-1,1,i_2i_1}^t \quad \forall \ s \in S,
  \]
  \[
  x_{s,t}^t = s_{s-1,1,i_1i_2}^t + z_{s,1,i_2i_1}^t \quad \forall \ s \in S,
  \]
  \[
  z_{s,1,i_1i_2}^t, z_{s,1,i_2i_1}^t \in \{0,1\} \quad s = 1,\ldots,2(n-1)-2;
  \]

- let \( C_y^t(i_1i_2i_3) \), where \( i_1i_2i_3 \in S^t_3(y) \), denote the set of constraints:
  \[
  \sum_{s=1}^{2(n-1)-2} (z_{s,1,i_1i_2i_3}^t + z_{s,1,i_2i_3i_1}^t) = 1,
  \]
  \[
  x_{s,t}^t = s_{s,1,i_1i_2i_3}^t + z_{s-2,1,i_2i_3}^t \quad \forall \ s \in S,
  \]
  \[
  x_{s,t}^t = s_{s-1,1,i_1i_2i_3}^t + z_{s-1,1,i_2i_3}^t \quad \forall \ s \in S,
  \]
  \[
  x_{s,t}^t = s_{s-2,1,i_1i_2i_3}^t + z_{s,1,i_2i_3}^t \quad \forall \ s \in S,
  \]
  \[
  z_{s,1,i_1i_2i_3}^t, z_{s,1,i_2i_3i_1}^t \in \{0,1\} \quad s = 1,\ldots,2(n-1)-2.
  \]

Observe that for a given \( s \in S \) and a given pair \((i,j)\) in \( A \), the variable \( x_{s,t}^t \) appears exactly once among the above constraints.

Finally, for each \( t \in T \), let \( H_y^t \), where \( s \in S \), denote the constraint:

\[
2h_s^t \geq \sum_{i \in S^t_1(y)} z_{s+1,i}^t + \sum_{i_1i_2 \in S^t_2(y)} (z_{s+1,i_1i_2}^t + z_{s+1,i_2i_1}^t) + \sum_{i_1i_2i_3 \in S^t_3(y)} \sum_{s=1}^{2(n-1)} (z_{s+1,i_1i_2i_3}^t + z_{s+1,i_3i_2i_1}^t) + \sum_{i \in S^t_1(y)} z_{s-1,i}^t + \sum_{i_1i_2 \in S^t_2(y)} (z_{s-2,i_1i_2}^t + z_{s-2,i_2i_1}^t) + \sum_{i_1i_2i_3 \in S^t_3(y)} \sum_{s=1}^{2(n-1)} (z_{s-3,i_1i_2i_3}^t + z_{s-3,i_3i_2i_1}^t).
\]

The slave problem is as follows:

\[
(SP_y) \quad \begin{align*}
C_y^t(i) & \quad \forall \ t \in T, \forall \ i \in S^t_1(y), \\
C_y^t(i_1i_2) & \quad \forall \ t \in T, \forall \ i_1i_2 \in S^t_2(y), \\
C_y^t(i_1i_2i_3) & \quad \forall \ t \in T, \forall \ i_1i_2i_3 \in S^t_3(y), \\
H_y^t & \quad \forall \ t \in T, \forall \ s \in S, \\
(4) & \quad (10).
\end{align*}
\]
If \((SP_y)\) is infeasible, then there is no schedule that realizes \(y\) and the following Benders cut can be added to the master problem:

\[
\sum_{t \in T} \left( \sum_{i \in S_1^t(y)} y^t_i + \sum_{i_1 i_2 \in S_2^t(y)} y^t_{i_1 i_2} + \sum_{i_1 i_2 i_3 \in S_3^t(y)} y^t_{i_1 i_2 i_3} \right) \leq \sum_{t \in T} \left( |S_1^t(y)| + |S_2^t(y)| + |S_3^t(y)| \right) - 1.
\]

However, this cut is rather weak because it simply forbids setting all the \(y\) variables that appear on the left-hand side to 1. As suggested by Example 3.1, a stronger cut can be obtained by finding subsets \(S_1^t(y), S_2^t(y)\) and \(S_3^t(y)\), of \(S_1^t(y), S_2^t(y)\), and \(S_3^t(y)\), respectively, such that the following is infeasible:

\[
(SP_y) \quad C^t_y(i) \quad \forall \ t \in T, \ \forall \ i \in S_1^t(y), \\
C^t_y(i_1 i_2) \quad \forall \ t \in T, \ \forall \ i_1 i_2 \in S_2^t(y), \\
C^t_y(i_1 i_2 i_3) \quad \forall \ t \in T, \ \forall \ i_1 i_2 i_3 \in S_3^t(y), \\
H^t_s \quad \forall \ t \in T, \ \forall \ s \in S.
\]

Then the following Benders cut can be added to the master problem:

\[
\sum_{t \in T} \left( \sum_{i \in S_1^t(y)} y^t_i + \sum_{i_1 i_2 \in S_2^t(y)} y^t_{i_1 i_2} + \sum_{i_1 i_2 i_3 \in S_3^t(y)} y^t_{i_1 i_2 i_3} \right) \leq \sum_{t \in T} \left( |S_1^t(y)| + |S_2^t(y)| + |S_3^t(y)| \right) - 1.
\]

One simple way to obtain such subsets is to go through each element \(\phi\) in \(\bigcup_{t \in T} (S_1^t(y) \cup S_2^t(y) \cup S_3^t(y))\) and solve the slave problem with the constraints \(C^t_y(\phi)\) removed to to see if it is infeasible (cf. the deletion filter introduced by Chinneck and Dravnieks [5]). One could also use algorithms that come with some solvers for finding small or minimal infeasible subsets of constraints for this purpose as well—for example, the Conflict Refiner in CPLEX 10.2. In our implementation, we used the CPLEX Conflict Refiner only for small values of \(n\) (\(n \leq 14\)) and with a time limit. Once the Conflict Refiner times out, the simple approach takes over to further refine the cut. One reason we did not use the Conflict Refiner in CPLEX exclusively is the lack of control over the time limit on the testing of each individual candidate that could potentially be removed. In addition, to avoid spending too much time on cut refinement, we did not insist that the subsets \(S_1^t(y), S_2^t(y)\) and \(S_3^t(y)\) be minimal when the number of teams is larger than 8.

**Remarks.** There is a wealth of literature on the subject of finding small or minimal infeasible subsets of constraints. For a survey on the subject, see Chapter 6 in [4].

### 4 Computational results

The computations for each of the instances were carried out on a Linux workstation with an Intel Pentium 4 3.00 GHz processor and 1.5GB of RAM. Most of the computational efforts were devoted to the NL instances. The NFL instances were included to test the effectiveness of the method on instances with a larger number of teams. Using the method described in the previous section, we obtained improved lower bounds for the mTTP instances NL10 for \(n = 10, 12, 14, 16\) and NFL10 for \(n = 16, 18, 20, 22, 24\). Because of memory limitation, NFL instances with 26 teams
or more were not considered. Table 1 lists the improved lower bounds along with the previous best known lower bounds, all but one of which are given by the MNTLB.

One can readily obtain better lower bounds by allowing the computations to continue. However, we decided to stop the computations for the NL instances when the point of diminishing return seemed to have been reached. The lists of the generated Benders cuts for the NL instances and the C++ source for a computer program that generates the master problem are available at the author’s web page.

In our computational experiments on instances with at most 14 teams, we found that the number of Benders cuts needed to reach a certain lower bound was sensitive to the time-out value for cut refinement. Therefore, the values in the last two columns of the table and are not necessarily best possible.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Previous best LB</th>
<th>This paper</th>
<th># Benders cuts</th>
<th>CPU time (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL10</td>
<td>58,277</td>
<td>58,769</td>
<td>2,275</td>
<td>22.5</td>
</tr>
<tr>
<td>NL12</td>
<td>110,519</td>
<td>111,064</td>
<td>2,343</td>
<td>23.0</td>
</tr>
<tr>
<td>NL14</td>
<td>182,996</td>
<td>183,631</td>
<td>967</td>
<td>15.3</td>
</tr>
<tr>
<td>NL16</td>
<td>253,957</td>
<td>254,242</td>
<td>163</td>
<td>12.4</td>
</tr>
<tr>
<td>NFL16</td>
<td>228,251</td>
<td>228,446</td>
<td>54</td>
<td>5.2</td>
</tr>
<tr>
<td>NFL18</td>
<td>276,395</td>
<td>276,519</td>
<td>42</td>
<td>4.5</td>
</tr>
<tr>
<td>NFL20</td>
<td>$^a$316,721</td>
<td>316,727</td>
<td>34</td>
<td>5.2</td>
</tr>
<tr>
<td>NFL22</td>
<td>383,971</td>
<td>384,001</td>
<td>25</td>
<td>5.2</td>
</tr>
<tr>
<td>NFL24</td>
<td>434,576</td>
<td>434,598</td>
<td>13</td>
<td>3.5</td>
</tr>
</tbody>
</table>

$^a$ Same as the ILB

Table 1: Improved lower bounds for mTTP benchmark instances

Our method did not obtain improvement on the lower bounds for the Circular Distance instances. In addition, judging from the rate of improvement of the lower bounds for the instances NL6 and NL8, we expected that memory would run out before reaching optimality. In contrast, the method described in [3] solved NL6 in a matter of seconds and NL8 in 4 days.

5 Final remarks

In this paper, a method to improve the lower bound for the mirrored Traveling Tournament Problem is described. The method computed improved lower bounds for all the benchmark NL instances with at least 10 teams and NFL instances up to 24 teams. However, the method could not solve to optimality instances with 6 or 8 teams within a reasonable amount of time. One possible reason is that the master problem might be too weak. One way to improve the method is to define a stronger master problem that can still be solved efficiently. Another way to improve the method is to keep a library of Benders cuts obtained either analytically or computationally so that they can be recognized without having to solve slave problem.
References


