

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

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



New models for the Mirrored Traveling Tournament Problem

Marco Antonio Moreira de Carvalho a,*, Luiz Antonio Nogueira Lorena b

- ^a Ouro Preto Federal University, Campus Morro do Cruzeiro, Ouro Preto 35400-000, MG, Brazil
- ^b National Institute for Space Research, São José dos Campos 12227-010, SP, Brazil

ARTICLE INFO

Article history: Received 18 April 2011 Received in revised form 19 June 2012 Accepted 20 August 2012 Available online 31 August 2012

Keywords: Traveling Tournament Problem Sports scheduling Integer programming

ABSTRACT

The Mirrored Traveling Tournament Problem (mTTP) is a challenging combinatorial optimization problem which consists in generating a timetable for sports tournaments with two half series, what is equivalent to a double round-robin timetable problem. The distance traveled by the teams should be minimized in the final timetable, and a new objective is to minimize the longest distance traveled, named <code>MinMaxTTP</code>. It is proposed an integer programming formulation to the mTTP and two models with dynamic constraints to its solution. Both models are based on the detection of independent sets on conflict graphs, whose use has not been reported in the literature about the problem. Real data benchmarks from a baseball tournament are used in the experiments carried out.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Due to its importance, the scheduling of sports tournaments (sports scheduling) has an area of research of its own within operational research. In such tournaments, teams from different locations face each other, making necessary trips to perform the matches, which implies in operational costs and can also affect athletes' performance due to the suffered fatigue. Thus, it is of interest to the tournaments organization to consider these aspects during the matches scheduling. In double round-robin tournament type, in which we are concerned, all the teams play against each other twice in two half series, alternating home rights (i.e. the responsibility for organizing the match). In this type of tournament, all matches are scheduled in advance, since the composition of matches does not depend on the result of others. For a review of terminology, applications, specific constraints and methods of this area, we refer the reader to Rasmussen and Trick (2008).

The Traveling Tournament Problem (TTP) (Easton, Nemhauser, & Trick, 2001) consists of, given a set of n (an even number) teams and the matrix of distances between the team's hometowns, to schedule all matches of a tournament. Among the objectives, it is possible to minimize the distance traveled by the teams or minimize breaks of patterns of alternating home rights. In this paper we consider a new objective to the TTP: to minimize the longest distance traveled, which we called MinMaxTTP. This problem is considered an operational research "discomfort" (Trick, 2010):

though this is an easy problem to be stated, on the other hand it is very hard to solve even for small instances.

The tournaments are divided into two half series, these in turn are divided into rounds. The constraints of TTP are:

- 1. all teams play against all other at each half series;
- 2. each team plays only one match per round;
- 3. two matches cannot occur consecutively between the same teams (no repeaters); and
- 4. no team can have more than three consecutive matches at home (i.e., as the host) or away (i.e., as the visitor). In the version addressed in this paper, called Mirrored Traveling Tournament Problem (mTTP), the second half has the same sequence of matches in the first, however, the home rights are mirrored (i.e., inverted). Thus, we schedule only the first half, removing constraint 3 and adding the following constraint to the problem:
- 5. a match and its mirror cannot occur in the same half series.

Note that teams do not necessarily have to return to their hometowns after each match (except in cases in which constraint 4 applies), resulting in the need to minimize the distance traveled, an objective considered in this approach. Additionally, it is assumed that each team is in its hometown before the first round of the first half and after the last round of the second half. Thus, a solution to the mTTP can be seen as a set of tours to be traveled by teams during the tournament, beginning and ending at their hometowns.

In this work, the mTTP is formulated as an integer programming problem in which the concept of conflict graph is used as the basis for the development of two models. To the best of our knowledge,

^{*} Corresponding author. Tel.: +55 31 3559 1663; fax: +55 31 3559 1662. E-mail addresses: mamc@iceb.ufop.br (M.A.M. de Carvalho), lorena@lac.inpe.br (L.A.N. Lorena).

the use of such concept has not been reported in the literature about the problem.

The remainder of this paper is organized as follows: In the next section, a literature review is performed. Then, the new formulation, conceptualization and integer programming models for the mTTP are presented. In Section 4, computational experiments are detailed and results discussed. Finally, conclusions are drawn about the work and references are shown.

2. Literature review

Although the complexity of TTP has not been defined, one can see that this is a complex problem. Among the approaches described below, some rely on grid computing and parallelism. According to the experiments analyzed, it becomes clear that even with the use of such methods, computing the solution of problems of moderate dimensions still requires days of execution.

Instances often used as benchmarks in the literature about the problem are real data from the National League of Professional Baseball Clubs (*NL*) Super 14 Rugby League (*Super*), National Football League (*NFL*) and the Brazilian Soccer Championship. These instances have 4–32 teams, and can be found in http://mat.gsia.cmu.edu/TOURN/. Among the sets cited, the most discussed among the works below is *NL*. In the remainder of this paper instances of this set will be referred to as *NLn*, where *n* denotes the number of teams. The largest instance solved up to date has 10 teams, and to prove optimality it took a week of running time in a grid consisting of 120 processors (Trick, 2010).

In Easton et al. (2001), the TTP is defined and real data instances are presented for use as benchmarks. It is also proposed a method of constraint programming and integer programming. In preliminary experiments reported, instances *NL4* and *NL6* were solved, the first in a few seconds and the last in some hours in a computational environment not informed. No greater instance was solved by this method.

The use of Lagrangean relaxation is proposed in Benoist, Laburthe, and Rottembourg (2001), in which a subproblem is created for each team: the restricted traveling salesman problem. The relaxation is used in a collaborative scheme that also involves constraint programming for bounds strengthening and dynamic programming. In their experiments with the benchmarks, the approach was able to prove optimality for instances *NL4* and *NL6*, requiring 24 h of execution for the latter. For the *NL10* instance, after 24 h of running the gap obtained was 17.7%. The lower bound obtained shown to be effective.

Easton, Nemhauser, and Trick (2003) again address the TTP, this time through a parallelized implementation of the branch-and-price method, where the master problem is solved by column generation and the subproblems are solved by constraint programming. The instance *NL8* was solved in just over 4 days, using 20 processors, however, the constraint of no repeaters was not considered. It was proposed the *independent lower bound*, a lower bound that considers a traveling salesman problem for each team.

In circular TTP instances, all the teams venues are located on a circle, such that the distance between two teams a and b is $min \{a-b,b-a+n\}$, where n is the number of teams. Recently, Gschwind and Irnich (2011) showed that in these cases, the problem symmetry can be used to reduce the search space by a factor approximating 4n. Lower bounds for circular instances are proposed by Fujiwara, Imahori, Matsui, and Miyashiro (2006).

A hybrid heuristic of GRASP (Greedy Randomized Adaptive Search Procedure) and ILS (Iterated Local Search) is presented in Ribeiro (2007). The results showed that in some cases the mirrored solution found was better than the best known not mirrored solution at that moment, for circular distance instances. The average gap in relation to the not mirrored solutions was 17.1%. Another

important aspect is the execution time limited to 15 min, much lower than that reported by other methods. Four different parallelization strategies on a computational grid for this heuristic are proposed in Araújo, Boeres, Rebello, Ribeiro, and Urrutia (2007). Using the strategy of maintaining a poll of solutions, within hours of execution it was possible to obtain better solutions than those previously reported for the mTTP for instances with circular distances and for *NL16*. A grid with 82 computers distributed in four clusters was used.

Urrutia, Ribeiro, and Melo (2007), improved the *independent lower bound* considering the optimal solutions if the distances were constant between all pairs of cities. Good reduction rates were obtained for both TTP and mTTP. For instances of the *NL* set, only for instance *NL8* no reduction in the existing bound for the mTTP was achieved. Bounds not available for the *NFL* set were also obtained.

Cheung (2008), presents a two stage method, based on 1-factorizations on the complete graphs representation of the mTTP. With this method, it was possible to solve the *NL8* instance, using a single computer without parallelization after 3.7 days of execution. Although it was subsequently published, the integer programming formulation solved by branch-and-price proposed by Irnich (2010) is considered the first method to solve the instance *NL8* to proven optimality, after 12 h of execution in a single computer without parallelism. Additionally, several lower bounds for benchmarks were improved, among them, *NL10* and *NL12*.

The Benders decomposition approach is used to obtain lower bounds for the mTTP in Cheung (2009), which additionally proposes a mixed integer linear programming model. Better bounds for the instances *NL10* to *NL16* and *NFL16* to *NFL24* were found, requiring 3.5–22.5 days of execution.

An annotated bibliography on sports scheduling detailing definitions, methodologies and applications is provided by Kendall, Knust, Ribeiro, and Urrutia (2010). Among the problems addressed is the mTTP.

3. New formulation and models

Two new integer programming models for mTTP are presented. Both models only schedule the first half of a tournament – the second half is mirrored and taken into account in the costs. Also, they are based on independent sets detection in specific graphs that have characteristics of conflicts between its variables, however, they have different objectives. Such graphs, named *conflict graphs*, and the proposed models are presented below. Additionally, the ways in which distances are calculated in the models are shown, each serving a specific need.

3.1. Conflict graphs

In conflict graphs, vertices represent instances that are adjacent to other only if they are conflicting. In the case of TTP, vertices represent matches, which are adjacent if they cannot be scheduled for the same round because they have teams in common. Applications of conflict graphs in other problems include Bin Packing with Conflicting Objects (Muritiba, Iori, Malaguti, & Toth, 2009), Point-Feature Cartographic Label Placement, Manufacturer's Pallet Loading, Woodpulp Stowage and Daily Photograph Scheduling of an Earth Observation Satellite (Ribeiro, 2007).

For example, consider four teams – SPO, FLA, CRU and GRE – from four distinct Brazilian cities, respectively: São Paulo, Rio de Janeiro, Belo Horizonte and Rio Grande do Sul. All crossings totalize 12 ordered pairs (home, away) that define the matches. The conflict graph for this example is shown in Fig. 1.

According to the graph, matches SPO vs. FLA and GRE vs. CRU could be scheduled to the same round, since there is no conflict between these matches. The structure of the conflict graph, therefore,

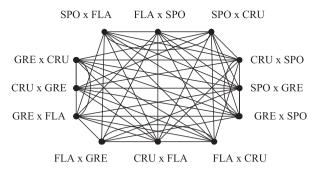


Fig. 1. Four teams' conflict graph. Adjacent matches cannot be scheduled to the same round.

suggests that independent sets (that in this case have a fixed size: n/2) represent matches that can be scheduled simultaneously for the same round. The mathematical models presented in the sections below are based on the detection of independent sets in conflict graphs, a new approach for the mTTP.

3.2. Problem data and variables

Both models use the same data set and share most of the variables. The problem data are:

- *n*: number of teams;
- p: number of rounds per half (i.e., n-1);
- m: number of possible ordered pairs of teams that make up a match (i.e., $n \times (n-1)$). Matches are ordered such that a match and its mirror are consecutive;
- *E*: edge set of the conflict graph;
- D: lower bound on the longest distance traveled by a team;
- g_i : distance between the cities of the teams participating in the match i (i = 1, ..., m), for cases in which only one of the teams will have to travel:
- $d_{i,k}$: distance to be traveled between matches i and k ((i,k) \in E) and consequently between its mirrors, as described below; and
- h_{i,k}: distance traveled between consecutive matches i and k
 ((i,k) ∈ E) of different halves (i is a last round match of the first
 half and k is a first round match of the second half).

The models variables are:

- x_{i,j}: takes value 1 if match i takes place in round j and value 0 otherwise (i = 1,...,m and j = 1,...,p);
- y_{i,k}: takes value 1 if match k takes place in the next round of the match i and value 0 otherwise ((i,k) ∈ E);
- w_{i,k}: takes value 1 if match i takes place in the last round of the first half and match k occurs in the first round of the second half; otherwise the variable takes value 0;
- s_i: takes value 1 if match i takes place in the first round of the first half and value 0 otherwise (i = 1,...,m);
- e_i: takes value 1 if match i takes place in the last round of the second half and value 0 otherwise (i = 1,...,m); and
- z: longest distance traveled during the tournament.

All variables are binary, except for *z*. As shown, calculating the distance to be traveled by a team during a tournament depends on the position of each one of its matches in the schedule and what matches are played at its home. Three different calculations of distances are used by the proposed models. Consider the match scheduling shown in Table 1 for the details below. The half and round of each match are shown, which are also identified by the value in column "id". The last two columns present the teams participating in each match and the city where it will take place.

The first distance calculation, denoted by g_i , is the distance between the hometowns of the two teams participating of match i. This distance is useful in cases where match i is a match from the first round of the first half or from last round of the second half (which is indicated by variables s_i and e_i in the model, respectively) and consequently only one team will have to travel. For example, according to Table 1, variable s_1 takes value 1 and g_1 is calculated as the distance between Porto Alegre and São Paulo, since the GRE team will travel to São Paulo. Similarly, variable e_{12} takes value 1 and g_{12} is calculated as the distance between São Paulo e Rio de Janeiro, since the FLA team will travel back to its hometown after the match.

Since it is the mirrored version of the problem, the first half determines the second, and therefore, only the first round variables are handled. However, the distance traveled in the second half must also be taken into consideration, for the integer programming model to represent the problem completely. From this need, a second distance table is created in which, for each pair of matches, the distances between them in the first half and between its mirrors in the second half are summed.

The calculation $d_{i,k}$ is the distance between two matches i and k with one team in common summed with the distance between its mirrored matches, that is, the distance to be traveled by one team between two matches in the first and second half. For example, $d_{3,6}$ is the sum of distances between São Paulo and Rio de Janeiro (the travel of SPO team between matches 3 and 6 of the first half) and between Belo Horizonte and São Paulo (the travel of SPO team between matches 9 and 12 of the second half – mirrors of the previous matches). If there is no team in common between matches, this distance is zero.

The third calculation, $h_{i,k}$ refers to the distance between matches i and k, where i is the last match of the first half and k is the first match of the second half. For example, $h_{6,7}$ is the distance between Rio de Janeiro and Porto Alegre.

3.3. Model 1

The objective of the first model is to minimize the total distance traveled between the scheduled matches, and is presented below.

Minimize
$$\sum_{(i,k)\in E} d_{i,k} y_{i,k} + \sum_{i=1}^{m} g_{i} s_{i} + \sum_{i=1}^{m} g_{i} e_{i} + \sum_{(i,k)\in E} h_{i,k} w_{i,k}$$
 (1)

Subject to:
$$\sum_{i=1}^{p} (x_{i,j} + x_{i+1,j}) = 1$$
 $i = 1, 3, 5, ..., m-1$ (2)

$$x_{i,j} + x_{k,j} \leqslant 1 \qquad (i,k) \in E \tag{3}$$

$$j = 1, ..., p$$
 $x_{i,j} + x_{k,j+1} - y_{i,k} \le 1$ $(i,k) \in E$ (4)
 $j = 1, ..., p - 1$

$$x_{i,1} + x_{i+1,1} - s_i = 0$$
 $i = 1, 3, 5, ..., m-1$ (5)

$$x_{i,p} + x_{i+1,p} - e_i = 0$$
 $i = 1, 3, 5, ..., m-1$ (6)

$$x_{i,1} + x_{k,p} - w_{k,j} \leqslant 1$$
 $(i,k) \in E$ $j = \text{match } i \text{ mirror}$ $i = 1, \dots, m$ (7)

$$x_{i,j} \in \{0,1\}$$
 $i = 1, ..., m$ $j = 1, ..., p$ (8)

$$y_{i,k} \in \{0,1\} \qquad (i,k) \in E \tag{9}$$

$$w_{i,k} \in \{0,1\}$$
 $(i,k) \in E$ (10)

$$s_i \in \{0, 1\} \qquad i = 1, \dots, m$$
 (11)

$$e_i \in \{0,1\} \qquad i = 1, \dots, m$$
 (12)

Each term of the objective function considers the four different distance calculations, respectively, the distance between two matches with one team in common, the distances between the hometowns

Table 1 Example of schedule.

Half	Round	id	Match	City
1st	1	1	SPO vs. GRE	São Paulo
		2	FLA vs. CRU	Rio de Janeiro
	2	3	SPO vs. CRU	São Paulo
		4	FLA vs. GRE	Rio de Janeiro
	3	5	CRU vs. GRE	Belo Horizonte
		6	FLA vs. SPO	Rio de Janeiro
2nd	4	7	GRE vs. SPO	Porto Alegre
		8	CRU vs. FLA	Belo Horizonte
	5	9	CRU vs. SPO	Belo Horizonte
		10	GRE vs. FLA	Porto Alegre
	6	11	GRE vs. CRU	Porto Alegre
		12	SPO vs. FLA	São Paulo

of two teams that make up a match (considered for matches of the first and final rounds) and the distance traveled between matches of the end of the first half and the start of the second half.

The first set of constraints (2) provides that if a match is scheduled, its mirror may not be in the same half either. The independent set detection formulation (3) does not allow conflicting matches to be scheduled for the same round. Constraints (4)–(7) identify the type of each trip. If two consecutive matches have a team in common in the same half, variable y_{ik} takes value 1 (4). If a match or its mirror takes place in the first round, variable s_i takes value 1 (5). If a match or its mirror takes place in the final round, variable e_i takes value 1 (6). If a team plays matches k and j (where k is the last match of the first half and *j* is the first match of the second half), variable $w_{k,i}$ takes value 1 (7). As noted in the previous section, each one of these cases requires a different calculation of the traveled distance. These distances are summed in the objective function (1). Constraints (8)–(12) define that the variables are binary.

This model has $m \times (16 \times n + p + 30)$ variables and $m \times [1.5 + 1.5]$ $p \times (6 \times n - 12)$] constraints.

3.4. Model 2

The second model has a different objective: to minimize the longest distance traveled, which we named MinMaxTTP. To the best of our knowledge, this objective is not addressed in the literature about the TTP. The model is presented below.

Minimize
$$z$$
 (13)

Subject to:
$$\sum_{j=1}^{p} (x_{i,j} + x_{i+1,j}) = 1 i = 1, 3, 5, \dots, m-1 (14)$$
$$x_{i,j} + x_{k,j} \le 1 (i,k) \in E j = 1, \dots, p (15)$$
$$x_{i,j} + x_{k,j+1} - y_{i,k} \le 1 (i,k) \in E j = 1, \dots, p-1 (16)$$

$$x_{i,i} + x_{k,i} \leqslant 1 \qquad (i,k) \in E \qquad j = 1, \dots, p \tag{15}$$

$$x_{i,j} + x_{k,j+1} - y_{i,k} \le 1$$
 $(i,k) \in E$ $j = 1, \dots, p-1$ (16)

$$(D - d_{i,k})y_{i,k} + z \geqslant D \qquad (i,k) \in E$$

$$\tag{17}$$

$$x_{i,1} + x_{i+1,1} - s_i = 0$$
 $i = 1, 3, 5, \dots, m-1$ (18)

$$(D-g_i)s_i + z \geqslant D$$
 $i = 1, 3, 5, ..., m-1$ (19)

$$x_{i,p} + x_{i+1,p} - e_i = 0$$
 $i = 1, 3, 5, ..., m-1$ (20)

$$(D - g_i)e_i + z \geqslant D$$
 $i = 1, 3, 5, ..., m - 1$ (21)

$$X_{i,1} + X_{k,p} - W_{k,i} \le 1$$
 $(i,k) \in E$ (22)

i = match i mirror

$$i = 1, \ldots, m$$

$$(D - h_{i,k})w_{i,k} + z \geqslant D \qquad (i,k) \in E$$
 (23)

$$x_{i,j} \in \{0,1\}$$
 $i = 1,...,m, j = 1,...,p$ (24)

$$y_{i,k} \in \{0,1\} \qquad (i,k) \in E$$
 (25)

$$w_{i,k} \in \{0,1\} \qquad (i,k) \in E$$
 (26)

$$s_i \in \{0,1\} \qquad i = 1, \dots, m$$
 (27)

$$e_i \in \{0,1\} \qquad i = 1, \dots, m$$
 (28)

$$z \geqslant 0 \qquad (i,k) \in E \tag{29}$$

As in the previous model, constraints (14) and (15) impose that a match and its mirror cannot be scheduled both in the same half and that conflicting matches cannot be scheduled for the same round, respectively.

Constraints (16), (18), (20) and (22) identify the type of each trip performed similarly to the previous model. If two consecutive matches have a team in common in the same half, variable $y_{i,k}$ takes value 1 (16). If a match or its mirror takes place in the first round, variable s_i takes value 1 (18). If a match or its mirror takes place in the final round, variable e_i takes value 1 (20). If a team plays matches k and j (where k is the last match of the first half and j is the first match of the second half), variable $w_{k,i}$ takes value 1 (22).

The different distances are considered, and variable z (minimized in the objective function (13)) is forced be greater or equal the longest distance traveled by any of the teams, as follows:

- Constraint (17) forces z to be greater or equal than the distance traveled between two consecutive matches (and its mirrors) any team plays;
- Constraints (19) and (21) force z to be greater or equal than the distance traveled by any team in matches of the first or final rounds, respectively; and
- Constraint (23) forces z to be greater or equal than the distance traveled between two consecutive matches in different halves any team plays.

Finally, constraints (24)–(28) define that the variables are binary and the constraint (29) defines the variable is positive.

This model has $m \times (16 \times n + p + 30) + 1$ variables and $m \times [-24.5 + 16 \times n + (n-2) \times (6 \times p)] + 1$ constraints.

3.5. Consecutive home-away matches dynamic constraints

In the presented models, the constraint of consecutive home or away matches for each team was not directly considered. This constraint is applied dynamically to the model iteratively until we obtain a feasible solution to the problem.

At each loop, the model solution is analyzed and, if some team has more than three consecutive home or away matches, the variables related to such matches are embedded in a constraint as shown below:

$$x_{a,j}+x_{b,j+1}+x_{c,j+2}+x_{f,j+3}\leqslant 3$$
 $j=1,\ldots,p-3$ (30) $a,b,c,f=$ matches that violate the constraint

Once the constraint is added, the model is solved again and the process repeats iteratively until the solution meets all constraints. This method of applying the constraint prevents the integer programming model from becoming too large, adding only the necessary constraints.

As only variables related to the first half are manipulated (since the second half is mirrored), there still may be a violation of this constraint involving the two halves at the same time. For example, a team has two matches away in the final two rounds of the first half and again two matches away in the first two rounds of the second. It is also necessary to verify such special cases, what is done simulating the mirrors of the first rounds and confronting them with the latest rounds of the first half.

4. Computational experiments

Experiments were carried out involving some of the instances available in the literature described earlier in Section 2. The computing environment used consists of an Intel Pentium 4 3.20 GHz frequency and 1016 MB of RAM memory under Windows XP Professional Edition operational system. The models were solved using CPLEX 12.

In the first part of the experiment the models were solved directly, and then strategies for accelerating the solution were included. A limit of 48 h for the execution time was established. This limit is smaller than those seen in other works that address the TTP by exact methods, however, the results reported here are preliminary. Table 2 presents the data obtained by direct solution of the original models by CPLEX. For each model, the results, running times and number of loops required to solve the instances are presented. The last column of the table shows the best results known for each instance.

The direct solution of the models proves to be impossible within the time limit set, since in this condition only the smallest instance could be solved. In both models, such instance, *NL4*, was solved in less than a minute, while for the instance with two more teams, *NL6*, the 48 h of execution was exceeded. The fast growth of execution times shows the great difficulty in solving the problem.

In order to accelerate the solve-and-constrain loops, two strategies are proposed. The first aims to interrupt the CPLEX execution after obtaining the first feasible solution, instead of waiting for the complete solution. Table 3 presents the results obtained by applying this strategy.

For the first three instances, the results were similar for the two models, which also required the same number of iterations and generated exactly the same schedule. This fact in models of different objectives indicates that there is a difficulty in moving among different solutions, also evidenced by the fact that a change in one match makes the whole schedule infeasible. For the last instance, model 1 obtained a better solution in less time and fewer iterations. The deterioration of the solutions when compared to the previous approach solutions was expected, since CPLEX was

Table 2 Original models results.

Instance Model 1 Result	Model 1	Model 1			Model 2		
	Runtime (s)	# Loops	Result	Runtime (s)	# Loops		
NL4	8276	0.36	0 ^a	8429	0.48	0 ^a	8276 ^b
NL6	c	c	c	c	c	c	26,588
NL8	c	c	c	c	c	c	41,928
NL10	с	c	c	с	c	c	63,832

a Not applied.

 Table 3

 Results interrupting the execution of CPLEX on the first feasible solution found.

Instance Model Result	Model 1			Model 2			Benchmark
	Result	Runtime (s)	# Loops	Result	Runtime (s)	# Loops	
NL4	8413	0.08	0 ^a	8413	0.03	0 ^a	8276 ^b
NL6	30,555	0.63	4	30,555	0.92	4	26,588
NL8	56,599	55.16	29	56,599	34.09	29	41,928
NL10	88,556	145.76	208	91,219	3458.25	685	63,832

^a Not applied.

Table 4Results interrupting the CPLEX execution accordingly to the gap.

Instance Model 1 Result	Model 1	Model 1			Model 2		
	Runtime (s)	# Loops	Result	Runtime (s)	# Loops		
Gap: 95%							
NL4	8569	0.05	0^a	8413	0.03	0^a	8276 ^b
NL6	31,068	3.59	8	30,555	0.95	4	26,588
NL8	55,955	635.08	69	56,599	34.08	29	41,928
NL10	88,556	122.86	208	91,219	3418.22	685	63,832
Gap: 90%							
NL4	8569	0.03	0^a	8413	0.03	0^a	8276 ^b
NL6	28,514	5.23	8	30,555	2.27	4	26,588
NL8	c	с	c	56,599	436.36	29	41,928
NL10	c	c	c	91,219	123057.31	685	63,832
Gap: 50%							
NL4	8276	1.00	0^a	8596	0.47	0^a	8276 ^b
NL6	c	c	c	c	c	c	26,588
NL8	c	c	c	c	c	c	41,928
NL10	c	c	с	c	c	c	63,832

^a Not applied.

^b Optimal solution.

^c Solution not found within time limit.

^b Optimal solution.

^b Optimal solution.

^c Solution not found within time limit.

Table 5Number of dynamic constraints added to the models per instance.

Instance	Model 1			Model 2		
	Feasibility	Gap 95%	Gap 90%	Feasibility	Gap 95%	Gap 90%
NL6	16	35	39	16	16	16
NL8	151	475	a	151	151	151
NL10	1658	1658	a	5190	5190	5190

^a Solution not found within time limit.

interrupted, however, this deterioration may have been outstanding, given the distance of over 38% when compared to the benchmarks.

The second acceleration strategy controls the interruption of the model solution according to the gap between the current solution and the bound used by CPLEX. Therefore, it is possible to control the solution deterioration level and run time growth. Table 4 presents the results for different gap values.

Even for a small variation of gap values (from 95% to 90%), the execution time grows very fast. For a gap set at 95%, model 1 has the worst solutions, execution times and larger number of iterations compared to the previous strategy, except for instance *NL10*, for which there was only a small change of the runtime. For a gap of 90% the model achieved improved solutions for *NL6* instance, with little variation in execution time, but was not able to solve larger instances within the time limit.

Model 2 indicates again a great difficulty in moving among solutions, obtaining a single solution both when the gap is set at 95% or 90%, with a very high difference in execution times.

Lowering the gap to 50% it is possible to solve only the smallest instance in both models, reaching its optimum value in a second for model 1 and a worse value for the model 2. However, the objective of the latter model is to minimize the longest distance traveled, the introduced *MinMaxTTP*, which in fact occurs for this instance, but affects the total distance traveled.

The comparison of the number of dynamic constraints added to each model is performed in Table 5. For each model, the values are presented according to the CPLEX interruption strategy. The *NL4* instance does not appear in the table, since it has 4 teams and no possible solution can violate the constraint of three home or away consecutive matches.

Model 1 adds a different number of dynamic constraints in accordance with the strategy used (except for instance *NL10*). This number increases as the gap is reduced, indicating that different solutions are visited, requiring greater effort to obtain better solutions. Conversely, model 2 remains constant the number of constraints added, suggesting that the same solutions are generated, regardless of the strategy used.

Comparing the two models, one can see that there is an alternation in obtaining the best results and execution times according to the context used, so that we cannot determine the dominance of one over the other.

According to the data presented, the great conflict between solution quality and run times is made evident – with little difference between the different criteria established for speeding up execution times – as well as the difficulty in moving between solutions.

5. Concluding remarks

The Traveling Tournament Problem (TTP) has practical applications in scheduling sports and has turned into an investigation area of its own within the operations research community (Rasmussen & Trick, 2008). In relation to the classical problems of the field, the TTP was defined recently, and attracts attention for being a difficult

problem to solve, although it can be easily stated. The objectives vary from minimizing the distance traveled by the teams during the tournament and minimizing breaks on travel patterns and matches as host. The variation addressed in this paper focuses on cases where the tournaments are divided into two similar half series, where the same matches occur, but inverted in relation to the host team. This variant is called the Mirrored Traveling Tournament Problem (mTTP).

An integer programming formulation for mTTP based on the detection of independent sets in conflict graphs where the vertices represent the possible matches between teams and adjacencies indicate that two matches cannot be scheduled for the same round was presented. To the best of our knowledge, this concept had not been reported in works related to the Traveling Tournament Problem.

Two models were proposed, which share the same conceptual basis, however, they have different objectives: the first seeks to minimize the sum of the distances traveled, and the second seeks to minimize the longest distance traveled, which we called *Min-MaxTTP*. The latter, not found in other studies. Both models use dynamic constraints to control the excess of consecutive home or away matches, a constraint of the problem. Thus, the solution was performed iteratively: the problem is solved, dynamically constrained and solved again until we obtain a final feasible solution.

In preliminary computational experiments involving benchmarks of the literature, the original models were solved directly by the solver used, and later, strategies to accelerate the iteractive process were included. The first strategy was to interrupt the solver after obtaining a feasible solution to the model used, and the second was to stop the solver according to the gap between the current solution and the expected solution. The data collected during the experiments indicate the difficulty of addressing the problem, occurring little difference in results obtained by different solution strategies. The execution time grows very quickly while the sizes of the instances or the quality of the solutions grow slowly.

Future work involves the inclusion of cuts in the proposed models to accelerate the solution and experiments with time limit of execution similar to those found in the literature, as well as implementation of metaheuristics for supplementary results. Bounds based on a Lagrangian relaxation-type with decomposition into clusters will also be subject of future research. Finally, the new objective of minimizing the largest distance traveled (here called *MinMaxTTP*) should be further explored.

Acknowledgements

This work was funded by the State of São Paulo Research Foundation – FAPESP, process 2009/51831-9 (first author), and partially funded by the National Counsel of Technological and Scientific Development – CNPq, processes 471837/2008-3 and 300692/2009-9 (second author).

References

Araújo, A. P., Boeres, C., Rebello, V. E., Ribeiro, C. C., & Urrutia, S. (2007). Exploring grid implementations of parallel cooperative metaheuristics. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R. F. Hartl, & M. Reimann (Eds.). Metaheuristics. oper. res./comput. sci. interfaces series (39, pp. 297–322). US: Springer.

Benoist, T., Laburthe, F., & Rottembourg, B. (2001). Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems. In: CP-AI-OR'2001, Wye College (pp. 15–26).

Cheung, K. K. H. (2008). Solving mirrored traveling tournament problem benchmark instances with eight teams. *Discrete Optimization*, 5(1), 138–143.

Cheung, K. K. H. (2009). A Benders approach for computing lower bounds for the mirrored traveling tournament problem. Discrete Optimization, 6(2), 189–196.

Easton, K., Nemhauser, G., & Trick, M. (2003). Solving the travelling tournament problem: A combined integer programming and constraint programming

- approach. In Pract. and theor. of autom. timetabling IV (PATAT'02). Lect. notes comput. sci. (Vol. 2740, pp. 100-109). Berlin, Heidelberg: Springer.
- Easton, K., Nemhauser, G. L., & Trick, M. A. (2001). The traveling tournament problem description and benchmarks. In T. Walsh (Ed.). Lect. notes comput. sci (CP) (Vol. 2239, pp. 580–584). Springer. Fujiwara, N., Imahori, S., Matsui, T., & Miyashiro, R. (2006). Constructive algorithms
- for the constant distance traveling tournament problem. In E. K. Burke & H. Rudov (Eds.), Proceedings of the pract. and theor. of autom. timetabling VI (PATAT'06) (pp. 135–146). Berlin, Heidelberg: Springer-Verlag.
- Gschwind, T., & Irnich, S. (2011). A note on symmetry reduction for circular traveling tournament problems. European Journal of Operational Research, 210(2), 452-456. http://ideas.repec.org/a/eee/ejores/v210y2011i2p452- 456.html>.
- Irnich, S. (2010). A new branch-and-price algorithm for the traveling tournament problem. European Journal of Operational Research, 204(2), 218-228.

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

 Muritiba, A. E. F., Iori, M., Malaguti, E., & Toth, P. (2009). Algorithms for the bin packing problem with conflicts. Informs Journal on Computing. http://dx.doi.org/ 10.1287/ijoc.1090.0355.
- Rasmussen, R. V., & Trick, M. A. (2008). Round robin scheduling A survey. European Journal of Operational Research, 188(3), 617-636.
- Ribeiro, G. M. (2007). Relaxação lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos (in portuguese). Master's thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos.
- Trick, M. (2010). Operations research: Embarrassments and the traveling tournament problem. Electronic. http://mat.tepper.cmu.edu/blog/?p=1020>.
- Urrutia, S., Ribeiro, C. C., & Melo, R. A. (2007). A new lower bound to the traveling tournament problem. In Proceedings of the IEEE Symp. on Comput. Intell. in Sched. Honolulu (pp. 15-18).