A Composite-Neighborhood Tabu Search Approach to the Traveling Tournament Problem*

Luca di Gaspero and Andrea Schaerf
Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica
Università di Udine
via delle Scienze 208, I-33100, Udine, Italy
email: {l.digaspero,schaerf}@uniud.it

May 8, 2006

Abstract

The Traveling Tournament Problem (TTP) is a combinatorial problem that combines features from the traveling salesman problem and the tournament scheduling problem. We propose a family of tabu search solvers for the solution of TTP that make use of complex combination of many neighborhood structures. The different neighborhoods have been thoroughly analyzed and experimentally compared. We evaluate the solvers on three sets of publicly available benchmarks and we show a comparison of their outcomes with previous results presented in the literature. The results show that our algorithm is competitive with those in the literature.

Keywords: Tabu search, Local search, Composite neighborhood, Traveling tournament, Tournament scheduling, Traveling salesman

Introduction

The Traveling Tournament Problem (TTP) has been proposed by Easton *et al.* (2001) and consists in scheduling a double round-robin tournament, while minimizing the total traveling distance of the teams and satisfying a set of constraints. TTP is considered an interesting problem from both the theoretical and the practical side. On the one hand, it has a natural combination of features from the traveling salesman problem (Jünger *et al.*, 1995) and the tournament scheduling problem (Henz, 2001; Schaerf, 1999), so that it exhibits strong feasibility issues (due to the tournament structure) together with a complex optimization part (traveling distance). On the other hand, sport scheduling problems are important for their economical value. In fact, National sport leagues, mainly in Western Europe and USA, represent a big business whose profits also depend on the quality of schedules (see Easton *et al.*, 2001, for a discussion).

Due to the mentioned combinations of features, TTP has shown to be difficult to solve by means of exact methods even for very small instances. For example, its solution by either Integer Programming or Constraint Programming so far has been possible only for instances up to eight teams (Easton *et al.*, 2003). A set of challenging instances for which many researchers have provided solutions, but in many cases the optimal one is still to be found, is available

*To appear in *Journal of Heuristics*. The original publication will be available at www.springerlink.com

	Team	ATL	NYM	PHI	MON	FLA	PIT
0	ATL	0	745	665	929	605	521
1	NYM	745	0	80	337	1090	315
2	PHI	665	80	0	380	1020	257
3	MON	929	337	380	0	1380	408
4	FLA	605	1090	1020	1380	0	1010
5	PIT	521	315	257	408	1010	0

Figure 1: The NL6 instance

at the TTP webpage, maintained by Trick (2005). TTP is thus an interesting and promising problem for heuristic and meta-heuristic search approaches.

In fact, using simulated annealing Anagnostopoulos *et al.* (2005) have been able to find the best known solutions for many instances of the most popular testbed. Using a two-stage approach, also based on some local search techniques, Lim *et al.* (2005) have been able to improve some of the result of Anagnostopoulos *et al.* In addition, Ribeiro and Urrutia (2004) and Urrutia and Ribeiro (2005), still using a local search technique, have obtained very good results for the mirrored version of TTP (explained below). Finally, some of the best results have been obtained by Langford (2005) using a modified version of the simulated annealing of Anagnostopoulos *et al.*

In this work, we tackle the TTP using an approach based on tabu search and on the combination of neighborhoods. Our best solver is competitive with those in the literature, as shown by the results in Section 4.

1 The Traveling Tournament Problem

The input of TTP is an (even) positive integer n and an $n \times n$ distance matrix. The output is a double round-robin tournament on n teams; that is, a tournament with $r = 2n - 2$ rounds such that each team plays one game in every round, and each team plays each other team exactly twice: once home and once away.

A solution to the TTP problem must satisfy the following two (hard) constraint types:

H1, No repeat: A match between two teams can never be followed in the next round by the other match between the same teams.

H2, At most: A team cannot play more than 3 consecutive games at home or away.

The objective of the problem is to minimize the total traveling distance, assuming that a team starts the tournament at home and returns home at the end of the tournament (but not between two away games). As an example, consider the instance called NL6, which has $n = 6$ teams and the distance matrix reported in Figure 1. For this instance the optimal cost is known (23916) and the solution shown in Figure 2 is an optimal one.

The solution matrix must be interpreted as follows. The sign represents the location of the match: '+' home, '-' away; the number is the opponent index. For example, the games of the first round are: 4-0, 1-2, and 3-5 (FLA-ATL NYM-PHI MON-PIT). The last column is the total traveling distance of each team. For example team 0 (ATL) goes at FLA, then at NYM, then home, stays home, goes to PIT, and so on. The total distance traveled by this team is: 605 + 1090 + 745 + 0 + 521 + 408 + 380 + 665 + 0 + 0 + 0 = 4414.

		Round										Total
No.	Team	0	1	2	3	4	5	6	7	8	9	Distance
0	ATL	-4	-1	+3	+2	-5	-3	-2	+5	+1	+4	4414
1	NYM	+2	+0	+5	-3	-2	+4	+3	-4	-0	-5	3328
2	PHI	-1	+5	-4	-0	+1	-5	+0	+3	+4	-3	3724
3	MON	+5	-4	-0	+1	+4	+0	-1	-2	-5	+2	3996
4	FLA	+0	+3	+2	-5	-3	-1	+5	+1	-2	-0	5135
5	PIT	-3	-2	-1	+4	+0	+2	-4	-0	+3	+1	3319

Figure 2: An optimal solution to the NL6 instance

An additional constraint considered by Ribeiro and Urrutia (2004) (and imposed in many sport leagues) is the *mirror constraint*. The tournament must be composed by two legs, i.e., two consecutive single round-robin tournament parts which are identical except that home and away positions are inverted.

In this work, we do not impose the mirror constraint and we work in the general form of this problem.

2 Local Search for TTP

In order to apply local search to TTP, we have to define several features. First, we consider the search space, the cost function and the procedure for computing the initial state. Then, we define several possible structures of the neighborhood and we analyze their characteristics and landscape. Finally, we discuss compositions, variants and restrictions of the proposed neighborhood structures.

2.1 Search Space, Cost Function, and Initial State

As search space, we select the set of all correct tournament schedules, including those violating H1 and H2. That is, in any state it is not possible that a team plays zero or two games in a single round, but for example it is possible that two teams play twice back to back (violating H1).

The cost function is a weighted sum of the violations of H1 and H2 plus the travel distance. The weights are adjusted so that hard constraints are always more valuable than distance. However, as we will see below, the weights vary dynamically during search, so that it is possible to explore also the infeasible region of the search space.

The initial state is generated in two steps: First, we select a *tournament pattern*, i.e. a tournament in which "anonymous" numbers from 0 to $n-1$ are used as teams. Second, we randomly associate actual teams with distinct numbers in the pattern.

The problem of determining the tournament pattern is related to the problem of finding a *1-factorization* of a complete (undirected) graph (Mendelsohn and Rosa, 1985). Given the complete graph K_n (n even) we must partition it in a set of $n-1$ sets of $n/2$ arcs (called *1-factors*), such that in each set the arcs are pairwise non adjacent. Each arc represents a match and each 1-factor a round. Therefore, giving an order to the 1-factors and assigning home or away teams for each match, a 1-factorization can be turned into a tournament pattern.

The pattern we use is the so-called *canonical pattern* (see, e.g., de Werra, 1981), which can be easily computed in linear time. The first half of the canonical pattern for $n = 6$ is shown in Figure 3. The canonical pattern is mirrored, therefore the second half is the same with

Round 1	1-0	2-5	4-3
Round 2	0-2	3-1	5-4
Round 3	3-0	4-2	1-5
Round 4	0-4	5-3	2-1
Round 5	5-0	1-4	3-2

Figure 3: The canonical pattern for $n = 6$

home/away swapped. In order to obtain different initial states for each run, the order of rounds and home/away selection are randomized. After these operations, the outcome is a random non-mirrored pattern.

2.2 Neighborhood definition

Many different neighborhoods can be applied to the proposed search space. In this work we analyze five different ones, which are adapted from Anagnostopoulos *et al.* (2005) and Ribeiro and Urrutia (2004), with some modifications. We have experimented also with other neighborhood relations, but with no significant result so far.

Calling T the set $\{0, \ldots, n-1\}$ of teams, and R the set $\{0, \ldots, r-1\}$ of rounds, the neighborhoods are defined as follows:

N_1, SwapHomes:

Attributes: $\langle t_1, t_2 \rangle$, where $t_1, t_2 \in T$.
Preconditions: $t_1 \neq t_2$
Effects: Swap home/away position of the two games between t_1 and t_2.

N_2, SwapTeams:

Attributes: $\langle t_1, t_2 \rangle$, where $t_1, t_2 \in T$.
Preconditions: $t_1 \neq t_2$
Effects: Swap t_1 and t_2 throughout the whole state.

N_3, SwapRounds:

Attributes: $\langle r_1, r_2 \rangle$, where $r_1, r_2 \in R$.
Preconditions: $r_1 \neq r_2$
Effects: All matches assigned to r_1 are moved to r_2, and vice versa.

N_4, SwapMatches:

Attributes: $\langle r, t_1, t_2 \rangle$, where $r \in R$ and $t_1, t_2 \in T$.
Preconditions: $t_1 \neq t_2$, and t_1 and t_2 do not play together at r
Effects: The opponents of t_1 and t_2 in r are exchanged. The teams t_1 and t_2 remain in the same location (home or away), their opponents possibly change location.
Note: Needs extra changes to do not fall outside the search space (see below)

N_5: SwapMatchRound

Attributes: $\langle t, r_1, r_2 \rangle$, where $t \in T$ and $r_1, r_2 \in R$.

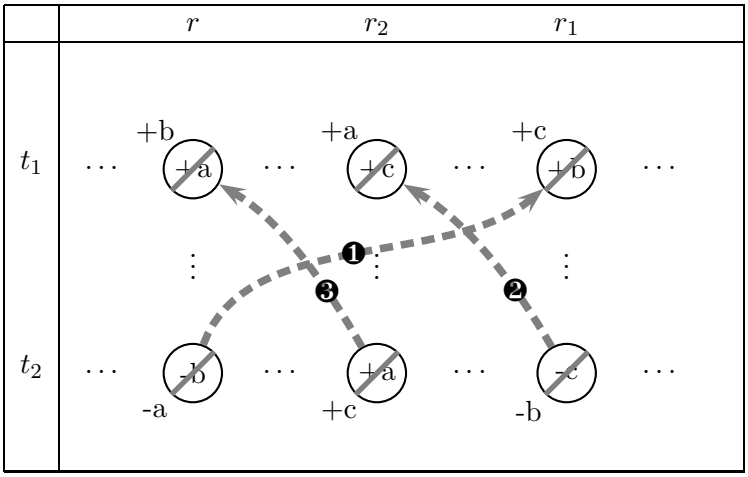


Figure 4: The construction of the repair chain of the move $N_4\langle r, t_1, t_2\rangle$

Preconditions: $r_1 \neq r_2$ and t doesn't play with the same opponent in r_1 and r_2

Effects: The opponents of t in rounds r_1 and r_2 are exchanged. The team t remains in the same location (home or away), its opponents possibly change location.

Note: Needs extra changes to do not fall outside the search space (see below)

It is easy to see that for both N_4 and N_5 the execution of a move leads outside the search space. In order to obtain a valid tournament structure the move must be followed by a set of other changes involving also other teams and rounds, which we call the *repair chain*.

There are different ways to obtain a valid tournament, and thus different ways to define the repair chain. We consider one viable option described in the following. Alternative definitions will be discussed in Section 2.4.

Regarding N_4, consider the move $m = N_4\langle r, t_1, t_2\rangle$ and the fragment of state sketched in Figure 4, where in round r there are the matches $t_1 - a$ and $b - t_2$. These matches are removed and replaced by the matches $t_1 - b$ and $a - t_2$. The repair chain starts at round r_1, where the match $t_1 - b$ takes place, and t_2 incidentally plays with the team c. The two matches $t_1 - b$ and $c - t_2$ are replaced by the matches $t_1 - c$ and $b - t_2$, and the chain continues on the round r_2 where $t_1 - c$ takes place. In the case of Figure 4, in round r_2 the match $t_2 - a$ takes place, and the chain closes up to round r. In this case, m has length 3, and the rounds included are r, r_1, r_2. In general a move length could vary from 2 up to $r - 2$. The upper limit is because the two rounds in which t_1 and t_2 play together are never touched by an N_4 move.

Regarding N_5, consider now the move $m = N_5\langle t, r_1, r_2\rangle$ and assume that the match $t - a$ takes place at r_1 (see the example in Figure 5). This match is moved to r_2 and a is inserted in the repair chain. Subsequently we search for the match of a in r_2, say $b - a$. This match is moved to r_1 and b is inserted in the repair chain. We continue by searching for the match of b in r_1 which is moved to r_2. The chain continues with the opponents of b in r_1 as long as we get back to t. In the extreme case, all teams are involved and the move collapses to the move $N_3\langle r_1, r_2\rangle$.

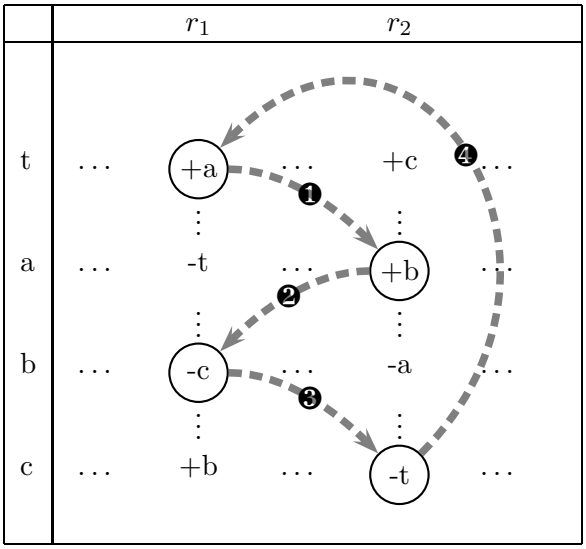


Figure 5: The construction of the repair chain of the move $N_5\langle t, r_1, r_2\rangle$

2.3 Neighborhood analysis

The natural candidate to be used as neighborhood in the meta-heuristics is the union of all basic ones, i.e. $\bigcup_{i=1}^{5} N_i$. However, in this section we analyze in deep the structure of N_i, in order to obtain more insights for the choice of the neighborhood for the solving meta-heuristics. Among others, we search for hints for reducing the size on the neighborhood, without loosing in the quality of the states explored.

First of all we notice that, due to the composite nature on the neighborhoods, it is possible that different moves do actually lead to the same state. This means that, calling M_i the set of *moves* that corresponds to the set of (neighbor) *states* N_i, there is the possibility that $\mid M_i \mid \; > \; \mid N_i \mid$. In a given state, two moves leading to the same state are said to be *equivalent (in that state).*

Detecting the existence of equivalent moves is an important issue for tabu search, which explores large parts of the full neighborhood at each iteration. Its importance is not only related to efficiency reasons, i.e. to avoid checking a state twice, but also to the effectiveness of the tabu mechanism. In fact, it may happen that a move should be tabu, but its tabu status is not detected because it is not actually in the tabu list (while an equivalent one is), thus leading to cycles in the search. This is not an issue for simulated annealing, which is based on a random move selection, and that is the reason why this issue is neglected by Anagnostopoulos *et al.* (2005), who make use of simulated annealing.

First, we notice that there are some trivial equivalences due to symmetries. For example, it is easy to see that in any state the move $N_1\langle t_1, t_2\rangle$ is equivalent to the move $N_1\langle t_2, t_1\rangle$. This and similar cases are easily dealt with by simply imposing symmetry breaking constraints on the move attributes, in this case by considering only moves such that $t_1 < t_2$.

Furthermore, we realize that, in a given state a move $N_5\langle t, r_1, r_2\rangle$, whose repair chain is composed by the teams $t_1, \ldots, t_k$, is equivalent to any of the moves $N_5\langle t_i, r_1, r_2\rangle$, with $1 \leq i \leq k$. In fact, it is easy to see that the repair chain of the move $N_5\langle t_i, r_1, r_2\rangle$ is composed by the same k teams $\{t_{i+1}, \ldots, t_k, t, t_1, \ldots, t_{i-1}\}$, so that the moves represent the same overall cycle, just started at a different point.

For example, in the instance NL6 shown before all the moves $N_5\langle t, 1, 4\rangle$, with $t \in \{0, 1, 2, 5\}$

Instance	$N_1 = M_1$	$N_2 = M_2$	$N_3 = M_3$	N_4	M_4	$N_5 = M_5$
NL10	45	45	153	109.5	117.5	195.2
NL12	66	66	231	166.4	175.8	309.2
NL14	91	91	325	250.8	264.8	470.4
NL16	120	120	435	341.7	358.4	652.5

Table 1: Cardinality of the neighborhoods (N_i) and the move sets (M_i)

lead from the state in Figure 6 [left] to the state shown in Figure 6 [right].

-4	-1	+3	+2	-5	-3	-2	+5	+1	+4
+2	+0	+5	-3	-2	+4	+3	-4	-0	-5
-1	+5	-4	-0	+1	-5	+0	+3	+4	-3
+5	-4	-0	+1	+4	+0	-1	-2	-5	+2
+0	+3	+2	-5	-3	-1	+5	+1	-2	-0
-3	-2	-1	+4	+0	+2	-4	-0	+3	+1

$\Longrightarrow$

-4	**-5**	+3	+2	**-1**	-3	-2	+5	+1	+4
+2	**-2**	+5	-3	**+0**	+4	+3	-4	-0	-5
-1	**+1**	-4	-0	**+5**	-5	+0	+3	+4	-3
+5	-4	-0	+1	+4	+0	-1	-2	-5	+2
+0	+3	+2	-5	-3	-1	+5	+1	-2	-0
-3	**+0**	-1	+4	**-2**	+2	-4	-0	+3	+1

Figure 6: State Transition for $N_5\langle t, 1, 4\rangle$ with $t \in \{0, 1, 2, 5\}$ (changes are in bold)

To provide against this situation, we generate only the moves $N_5\langle t, r_1, r_2\rangle$ such that the number t is smaller than all other teams in its repair chain. In the example, only the move $N_5\langle 0, 1, 4\rangle$ is generated whereas $N_5\langle 1, 1, 4\rangle$, $N_5\langle 2, 1, 4\rangle$, and $N_5\langle 5, 1, 4\rangle$ are discarded.

Analogously, a move $N_4\langle r, t_1, t_2\rangle$ is always equivalent to a move $N_4\langle r_1, t_1, t_2\rangle$ if r_1 is in the repair chain of the first one.

In order to detect other, less trivial, equivalences we perform a test run in which all states reached applying a move m are stored in an array, and the duplications in the array are checked and counted (trivial duplications have been previously removed). Since equivalence is state dependent, we perform this test starting from a set of 50 different states and we record the average values. The states selected are random local minima obtained from a simple steepest descent algorithm using the neighborhood $\bigcup_{i=1}^{5} N_i$ and starting from different random initial states.

Table 1 shows the results for benchmark instances of the family NLx (where x is the number of teams).

From Table 1, we see that for N_1, N_2, N_3, and N_5 there are no further move duplications. Conversely, N_4 has about 4-7% of moves equivalent to others (further discussed later).

We can also notice that N_1, N_2, and N_3 have fixed (quadratic) cardinality, respectively equal to $n(n-1)/2$, $n(n-1)/2$, and $r(r-1)/2$, independently of the state. Conversely, for N_4 and N_5 both the number of neighbors and the number of duplications (trivial or not) depend on the current state.

So far, we have dealt with the duplications inside a single neighborhood, but, as we will see, they may happen also between different ones. In other words, in a given state, it is possible that $N_i \cap N_j \neq \emptyset$, for $i \neq j$. Table 2 shows the average size of the intersections between neighborhoods (only non-zero ones) for the same set of 50 states.

Table 2 shows some interesting phenomena:

- The cardinality of $N_3 \cap N_5$ is significant, and also state dependent. This happens when the length of an N_5 move is equal to n.
- The cardinality of $N_2 \cap N_4$ instead is zero. This is due to the fact that a N_4 move involves

Instance	$N_3 \cap N_5$	(% of N_3)	(% of N_5)	$N_4 \cap N_5$	(% of N_4)	(% of N_5)
NL10	80.32	(52.5%)	(41.1%)	21.08	(19.2%)	(10.7%)
NL12	116.5	(50.4%)	(37.6%)	26.52	(15.9%)	(8.6%)
NL14	142.7	(43.9%)	(24.7%)	39.3	(15.6%)	(8.3%)
NL16	181	(41.6%)	(27.7%)	47.22	(13.8%)	(7.2%)

Table 2: Average cardinality of intersections of neighborhoods (only non-zero)

at most $r-2$ rounds (all but the two rounds in which the two teams play with each other), whereas a N_2 move involves all r rounds.

- The cardinality of $N_4 \cap N_5$ is also significant, and will be further investigated below.

Regarding the last point (and the duplications inside N_4 highlighted in Table 1), we realize that such duplications are related to the following specific situation. There are 4 teams, say a, b, c, d, such that the games $a-b$ and $c-d$ take place in one round, say r_1, and the games $a-c$ and $b-d$ take place in another round, say r_2. In this situation, the movement that brings $a-b$ and $c-d$ in r_2 and $a-c$ and $b-d$ in r_1, can be achieved with any of the moves $N_4\langle r_1, a, c\rangle$, $N_4\langle r_2, b, d\rangle$, and $N_5\langle a, r_1, r_2\rangle$.

We checked both analytically and experimentally that the kind of duplication highlighted in this example is the only possible one, and that all duplications are related to N_4 moves of length 2 and N_5 moves of length 4. This situation can be easily dealt with, when exploring the composite neighborhood, by discarding all N_4 moves with length 2. With this action, we remove all duplications, without loosing any reachable state.

Our next analysis regards the *quality* of moves; that is, the variation of cost due to the move. For simplicity, we focus on the objective function (without taking into account H1 and H2 violations), and Table 3 shows for each neighborhood the average distance variation and its standard deviation (in parentheses) for the same 50 selected states.

Inst.	N_1	N_2	N_3	N_4	N_5
NL10	1039.4 (1149.1)	5814.9 (3824.3)	4332.6 (2107.1)	3210.8 (2678.7)	3157.0 (2078.2)
NL12	1248.0 (1532.6)	8444.6 (5817.4)	6102.5 (2952.2)	4567.4 (3961.9)	4293.0 (2900.6)
NL14	1995.9 (2744.4)	18814.3 (17784.8)	11924.8 (5470.3)	9100.5 (9618.5)	7852.4 (5559.4)
NL16	2373.9 (3073.3)	26932.3 (24966.9)	16190.8 (6651.5)	12537.6 (13001.9)	10357.9 (7027.6)

Table 3: Cost of moves (Distance only) grouped by neighborhood

Table 3 shows that N_2 and N_3 have an average high cost, meaning that they produce a large modification in the current state that turns out to be too disruptive. Conversely, N_1 moves, due to their local effect have high quality and are candidate to be more often selected in the local search process.

For N_4 and N_5 we go into more details, and Figures 7 and Figures 8 show the distributions of the cost of the moves grouped by length represented by means of box-and-whiskers plots. That is, for each move length, reported on the x-axis, the graph shows the range of variation (the interval $[min\_cost, max\_cost]$), indicated by the dashed vertical line, and the frequency distribution of the solutions founded. The latter measure is expressed by means of a boxed area showing the range between the 1st and the 3rd quartile of the distribution (the so-called

interquartile range, that accounts for 50% of the total frequency). Finally, the horizontal line within the box indicates the median of the distribution.

Notice that most values are positive, and this is related to the fact that the selected states are local minima. Notice, however, that some points are under the 0 tick. This is due to the fact that we plot only the distance component (without H1 and H2), whereas this points are local minima for the overall cost function.

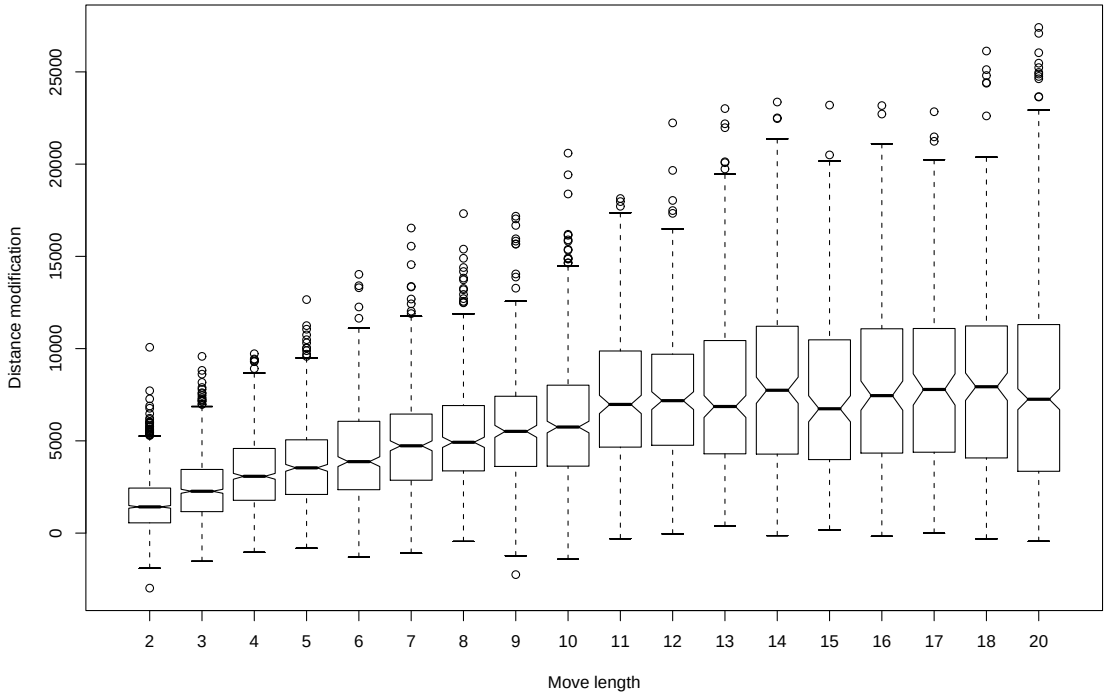


Figure 7: Costs for N_4 (Distance only) grouped by move length (instance NL12)

Figures 7 and 8 highlight that short moves clearly dominate longer ones. It also shows that for N_5 the average cost of the moves increases almost linearly with the size, whereas for N_4 from a certain size onward the cost decreases back. This latter phenomenon is intuitively explained with the fact that, the longer the move is, the more complete is the swap of two teams. A complete swap is generally less expensive than a "half-swap", because generally teams singularly have a coherent scheduling, and giving it to another team is less disruptive than breaking it between the two.

2.4 Neighborhood revision

Based on the observations of the previous section, we now devise a set of composite neighborhoods, called CN_i, that are candidates to give the best performances, and in Section 4 we compare them experimentally.

As the first neighborhood, we consider the set-union of all the basic ones, i.e.,

$$CN_1 = \bigcup_{i=1}^{5} N_i$$

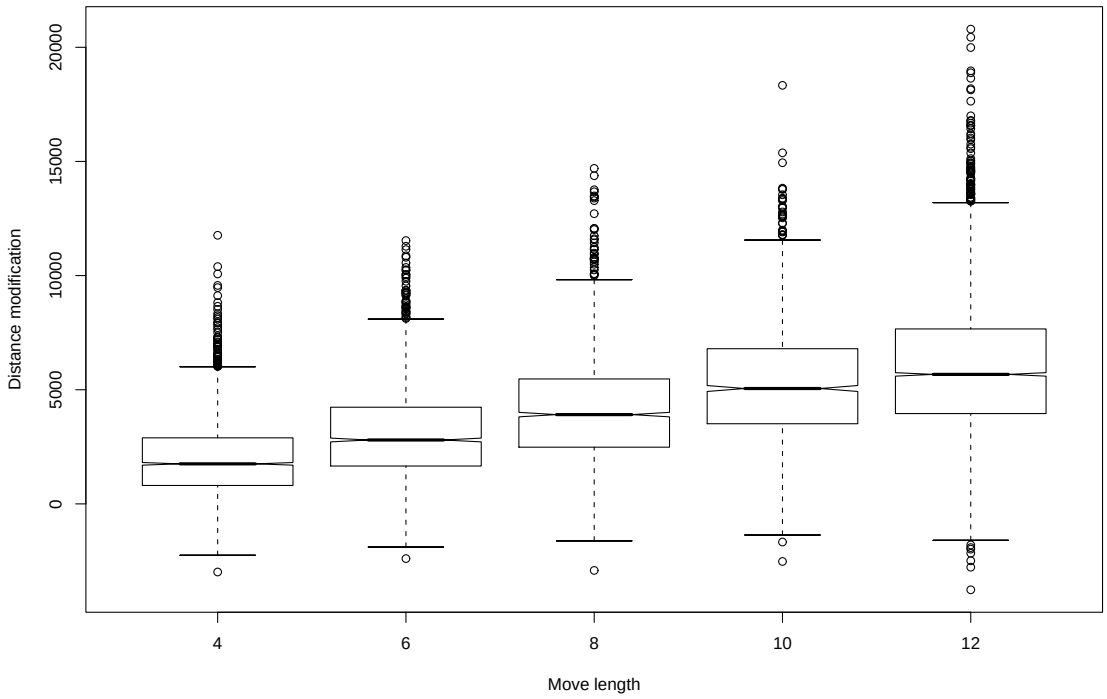


Figure 8: Costs for N_5 (Distance only) grouped by move length (instance NL12)

The next variant we consider is the possibility to exclude completely N_2 and N_3, given that they have shown bad scores. That is, we define

$$CN_2 = N_1 \cup N_4 \cup N_5$$

Next, we have seen that the best moves in N_4 and N_5 are those with short length. This fact suggests that exploring only moves with limited length could be an interesting way to reduce the size of the neighborhood, without affecting the quality of the solution. Looking at the distribution (and after experimentation) we fix the maximum length to 4 for N_4 and 6 for N_5. We therefore define

$$CN_3 = N_1 \cup N_4^{\leq 4} \cup N_5^{\leq 6}$$

where the superscript denotes the maximal length of the moves.

In this case, the construction of each candidate move is aborted as soon as its length goes above the maximum. This way we waste the time to generate the move (up to that length), but we save the time to evaluate it, which is significantly longer.

Obviously, the above ideas need to be verified experimentally, because larger (more perturbative) moves might still have a positive role in the overall meta-heuristic procedure, as suggested in many meta-heuristics, such as ILS (Lourenço *et al.*, 2002), VNS (Hansen and Mladenović, 1999), and LNS (Ahuja *et al.*, 2000). Nevertheless, our intuition in the case of TTP is that all moves, except N_1, are already somewhat "large moves", and the long ones may lead "too far" from the current state.

The fourth (and last) candidate neighborhood also comes from the idea of having shorter moves, and it emerges from a deeper look to the repair chain mechanism for N_4. In the current implementation, given the move $N_4\langle r, t_1, t_2\rangle$ and assuming that $t_1 - a$ takes place at r, the

chain closes at the round where $t_2 - a$ takes place. However, it is possible that the chain passes on the round where is $a - t_2$, which also could close the chain leading to a different feasible state. Based on this observation, we consider a new version of N_4 based on the *shortest chain closure* (SCC), which consists in closing the chain as soon as one of the two closing matches in encountered. The neighborhood we use is thus

$$CN_4 = N_1 \cup N_{4_{SCC}}^{\leq 4} \cup N_5^{\leq 6}$$

Due to the fact that more moves become short due to the SCC, the cardinality of $N_{4_{SCC}}^{\leq 4}$ is larger than the cardinality of $N_4^{\leq 4}$. We experimentally compute them, and it results that the cardinality of $N_{4_{SCC}}^{\leq 4}$ is about three times the cardinality of $N_4^{\leq 4}$. Therefore neighborhood CN_4 is significantly larger than CN_3.

2.5 Discussion

The primary question about a neighborhood is whether the search space is connected under it. This property would ensure that from any initial state it is theoretically possible to reach any final state using some local search technique.

In TTP, however, the size and the structure of the search space is still under investigation, see e.g. (Dinitz *et al.*, 1994; Lindner *et al.*, 1976). Therefore, at present, it is not possible to formally assess whether it is connected under our neighborhoods or not.

To shed some light on this issue, we performed an experiment on the instance NL8, for which there is some information of the search space and the optimal solution is known. For eight teams the number of non-isomorphic tournament patterns is only six, and these patterns are listed by Wallis *et al.* (1972). We ran our solvers from a set of six initial states built from the six different patterns. The outcome was that from all of them and for all CN_i we were able to reach the known optimal solution. This result supports (but obviously does not prove) the hypothesis that the search space is indeed connected under all CN_i

3 Solution Meta-Heuristic for TTP

The driving meta-heuristic employed in the solver of this work is tabu search (Glover and Laguna, 1997). Four different versions of the solver are generated using the neighborhoods CN_i (with $i = 1, \ldots, 4$), and they are compared with each other. The other characteristics of tabu search are common to all versions and are selected as follows:

Neighborhood sampling: The *full* neighborhood is traversed and all non-tabu neighbors are evaluated.

Tabu length: The tabu list is dynamic, so that each performed move remains in the list for a number of iterations randomly selected between t_{min} and t_{max}.

Prohibition: A move is tabu if a move from the same basic neighborhood with the same values for all attributes is in the list.

Aspiration criterion: The aspiration criterion is the standard one: a tabu move is accepted if it leads to a state that is better than the current best one.

Stop criteria: The stop criterion is based on the number of iterations since the last improvement (parameter *max idle iterations*). However, when the number of idle iterations reaches its maximum, the search is not stopped but it restarts from the best state found that far.

This procedure is finally stopped when it could not find any improvement for a given number of rounds (parameter *max idle rounds*) or when a given timeout is reached.

Other features: Similarly to Anagnostopoulos *et al.* (2005), our tabu search procedure makes also use of an adaptive modification of the weights for the violations of the constraints H1 and H2. Namely, the weight of each component is let to vary according to the so-called *shifting penalty* mechanism: if for a number k of consecutive iterations all constraints of that component are satisfied (resp. not satisfied), then the weight is divided (multiplied) by a factor α. This mechanism changes continuously the shape of the cost function in an adaptive way, thus causing tabu search to pass through infeasible states and visit states that have a different structure than the previously visited ones.

Needless to say, many other settings for the above parameters have been tested, but with inferior or comparable results. For example, the fixed-length tabu list has given inferior results for all possible values. Stronger prohibition mechanisms, based on single attribute, have also shown worse results.

4 Experimental Analysis

We carried out an extensive experimental analysis. In this section, we present the settings and the main results obtained.

4.1 Benchmark instances

We tested our algorithm on all instances available at the TTP webpage (Trick, 2005):

- the NLx family, based on real data of the US National Baseball League;
- the CIRCx family, artificial instances with the cities placed on a circle;
- the CONx family, with a constant distance matrix, i.e., all pair of cities have the same distance;
- the single BRA24 instance, based on real data of the Brazilian soccer championship.

Each family is composed by an instance for each even number of teams, the x stands for the dimension of the instance. Since for instances with 4, 6, and 8 teams we easily obtain the best known results (which are also certified optimal for NL4 and NL6), we decided to focus on instances with at least 10 teams.

4.2 Comparison methodology, parameter setting, and implementation

To the aim of comparing the solvers built upon the composite neighborhoods $CN_1, \ldots, CN_4$ identified in Section 2.4 we decided to analyze the algorithms by means of the RACE approach developed by Birattari (2004). This procedure has been developed for the purpose of selecting the configuration of a meta-heuristic by testing each candidate on a set of trials. The configurations that perform poorly are discarded and not tested anymore as soon as sufficient statistical evidence against them is collected.

This way, only the statistically proven "good" configurations continue the race, and the overall number of tests needed to find the best configuration(s) is limited. Each trial is performed on the same randomly chosen problem instance for all the remaining configurations and a rank-based statistical test is used to assess which of them are discarded.

Every solver is regarded as a race candidate and is granted with the same amount of CPU time (1 hour on an AMD Athlon 1.5GHz running Linux). This implies that a solver is stopped after one hour even if its stop criterion has not been met yet. Each trial is performed by all the remaining algorithms on the same instance and the same randomly generated initial solution.

The setting of the tabu search parameters has been identified by means of a preliminary set of experiments. The values employed in the comparison are the following: 5,000 max idle iterations, 20 max idle rounds, a dynamic tabu list whose length varies from $\frac{2}{3}n$ to n (where n is the size of the instance, i.e., the number of teams).

Since the distribution of the final solution values is expected to be not normal, among the statistical tests considered in the RACE approach, we are forced to employ the non-parametric Friedman test (see, e.g., Conover, 1999) for comparing the ranks of the candidate algorithms.

The algorithms have been entirely coded in C++ and compiled using the GNU C/C++ compiler (v. 3.3.3) under Linux (Kernel 2.6.8). The solvers make use of the framework EasyLocal++ (Di Gaspero and Schaerf, 2003) for the implementation of the local search algorithms and of the RACE package (Birattari, 2005) of the R language for statistical computing (R Development Core Team, 2005).

4.3 Neighborhood comparison

The solvers have been tested on all the families of instances employing the RACE procedure described in the previous section, and limiting the number of trials to 50. The parameter for accepting the alternative hypothesis of the Friedman test is set to $p < 0.01$.

The performances of the solvers built on the $CN_1, \ldots, CN_4$ neighborhoods are summarized in Figure 9 by means of box-and-whiskers plots.

Figure 9a shows the distribution of the ranks obtained by each solver at each step of the RACE procedure (i.e., the raw data used for the Friedman statistical test). It is clear from the picture that the best results are obtained with those algorithms employing the CN_3 and CN_4 neighborhoods, which result in a comparable median value. However, the distribution of ranks of CN_4 features more lower values than the one of CN_3 and indeed the RACE procedure designates CN_4 as the best performer.

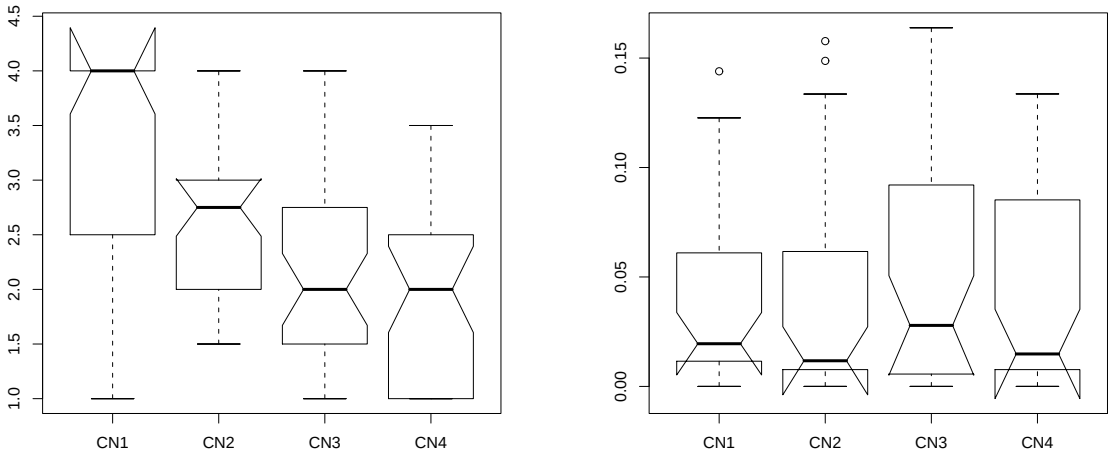
This tendency is confirmed also if we compare the algorithms with respect to their normalized costs (i.e., the ratio of the final cost value and the best known value for each instance) as shown in Figure 9b. However in this case the superiority of CN_4 is not as clear as in the previous plot. This is due to the different cost behavior of the algorithms on the various family of instances.

An example of the difference of cost behavior is reported in Figure 10. Figure 10a shows the distributions of costs for the NL12 instance for which the CN_4 solver is the clear winner. This is not the case on the CIRC14 instance (Figure 10b), for which CN_4 is in the same slot as CN_1 and CN_2.

4.4 Unlimited time experiments

For the purpose of comparing our best solver, i.e. the one with neighborhood CN_4, with previous results we performed a set of experiments with a larger number of idle rounds (namely 30) and without time limits.

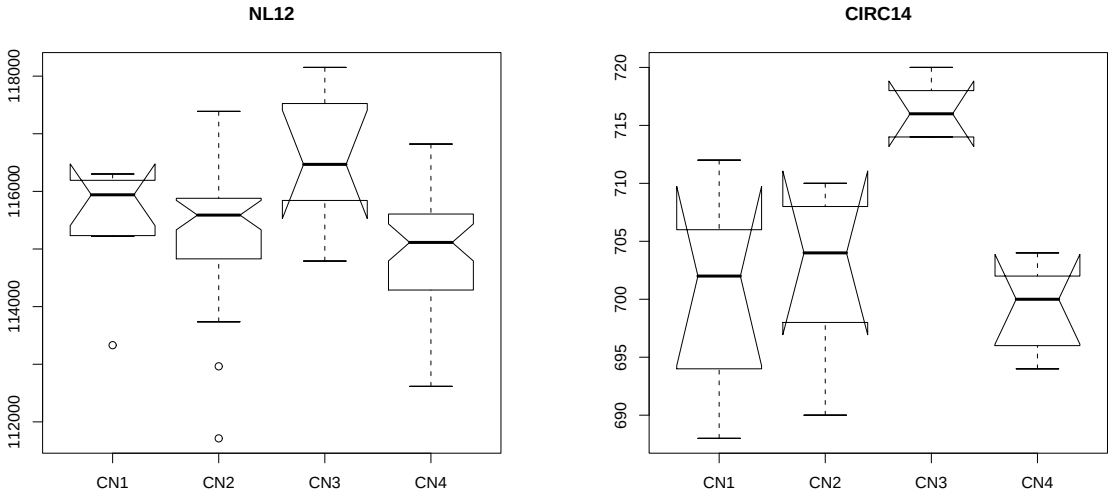
Table 4 shows the results obtained for all instances on 10 trials. The results show that the performances of the algorithm are relatively stable, and that the outcomes in terms of costs are comparable to those by (Lim *et al.*, 2005) and slightly worse than those by (Anagnostopoulos *et al.*, 2005). However, the running times are almost an order of magnitude lower with respect to both the other two, and on a comparable-speed machine.



(a) Distribution of the ranks

(b) Distribution of the normalized costs

Figure 9: Performances of the different TS based algorithms measured on all the instances



(a) Distribution of costs on the NL12 instance

(b) Distribution of costs on the CIRC14 instance

Figure 10: Examples of the cost behavior of the TS based algorithms on two specific instances

Instance	cost			time (secs)		
	min.	avg.	(std. dev.)	min.	avg.	(std. dev.)
NL10	59876	60424.2	(823.9)	3453.8	7056.7	(4453.9)
NL12	113729	114880.6	(948.2)	6480.9	10877.3	(5295.2)
NL14	194807	197284.2	(2698.5)	20657.8	29635.5	(7376.1)
NL16	275296	279465.8	(3242.4)	23242.3	51022.4	(19390.5)
CIRC10	256	259.2	(2.3)	3991.9	4446.7	(460.1)
CIRC12	438	440.4	(1.7)	10235.5	11597.4	(1449.0)
CIRC14	686	694.4	(6.4)	12729.7	14460.7	(2710.8)
CIRC16	1016	1030.0	(8.7)	19545.9	26833.6	(7849.7)
CIRC18	1426	1440.8	(10.5)	39149.0	55395.3	(16725.9)
CIRC20	1968	1998.4	(17.7)	59218.4	67189.8	(6053.8)
CON10	124	124.0	(0.0)	3364.9	3469.1	(130.9)
CON12	182	182.0	(0.0)	5906.9	6172.9	(295.0)
CON14	252	252.0	(0.0)	9761.7	11823.5	(1690.5)
CON16	329	329.2	(0.4)	14810.6	19665.4	(5350.8)
CON18	418	419.2	(0.8)	23803.2	33979.2	(9155.1)
CON20	522	523.0	(0.7)	39259.3	46579.2	(5232.5)
BRA24	530047	536648.8	(8081.6)	159086.0	246177.2	(119701.5)

Table 4: Results of the best solver

4.5 Best results

We now propose our *best* results, which are not coming from the systematic experimentation presented before, but from the overall set of experiments that we have conducted during this research (with various parameter settings and time limits). We compare them with respect to the best known ones.

Table 5 shows our best value, the current best together with the researcher(s) that obtained it. A number in bold means that our result is currently the best known. The * subscript means that the value has been proven to be optimal (by Rasmussen and Trick, 2005).

It is worth mentioning that the results of Araùjo *et al.* (2005) are all obtained in presence of the additional mirror constraint, therefore some of the best results for the general problem are currently mirrored solutions. Ribeiro and Urrutia obtained results also on the CONx instances with the mirror constraint, and they are described in (Urrutia and Ribeiro, 2005). They are however inferior to ours for the general (i.e., non mirrored) problem.

Instance	Our	Best	Obtained by
NL10	59,583	59,436	Langford
NL12	111,483	111,248	Anag. et al (2005)
NL14	190,174	189,766	Anag. et al (2005)
NL16	270,063	267,194	Anag. et al (2005)
CIRC10	242	242	Langford
CIRC12	426	408	Lim et al (2005)
CIRC14	668	654	Lim et al (2005)
CIRC16	1004	928	Lim et al (2005)
CIRC18	1412	1306	Araùjo et al (2005)
CIRC20	1946	1842	Lim et al (2005)

Instance	Our	Best	Obtained by
CON10	***124**	–	–
CON12	***181**	–	–
CON14	***252**	–	–
CON16	328	*327	Ras. & Tr. (2005)
CON18	**418**	–	–
CON20	**521**	–	–
CON22	**632**	–	–
CON24	**757**	–	–
BRA24	530043	503158	Araùjo et al (2005)

Table 5: Comparison with best results

Table 5 shows that our results are reasonably close to the best in the NLx instances; for

which they are, up to our knowledge, the second-best ones (for example, they are better than all the results of Lim *et al.*, 2005). For the CIRCx (except CIRC10) and the Brazilian instances, instead, we are still quite far from the best. Finally, they are the best for the CONx instances, although for them the only available results are the ones of Urrutia and Ribeiro (2005) who consider also the mirror constraint.

5 Conclusions and future work

We have proposed a tabu search approach for the TTP, focused on the definition of the neighborhood. The neighborhood employed in the search is a composition that has been designed starting from five basic neighborhood structures, according to an analytical study of the cardinality, the overlapping, and the average quality of the components.

The experimental evaluation has been conducted according to a sound statistical methodology, and it confirmed the intuitions arisen in the analytical study about the effectiveness of the different composite neighborhoods.

The experiments show that our best solver produce results comparable w.r.t. the best ones available in the literature in the publicly available instances.

For the future, we plan to work in the following directions:

- Design and compare new techniques. We aim at having an 'internal' comparison of the possible techniques. In details, we plan to implement at least simulated annealing and iterated local search, which, based on the literature on TTP, seem to be the most promising ones.

- Study further on the search space. As we mentioned above, the connectivity of the search space under the proposed neighborhood is still unclear. In our opinion, it is worth investigating theoretically whether this is the case, and possibly define a new search space. For example, a larger search space with teams that play more games in the same round (obviously taking it into account in the cost function of the meta-heuristic) could be an interesting alternative approach.

- Optimize the code. Due to the exhaustive exploration of the neighborhood, tabu search is intrinsically much slower than simulated annealing to perform each single iteration. In order to be competitive with the other techniques, tabu search needs to be implemented efficiently. In our case, many points still need to be improved, especially regarding the computation of the cost difference of two neighbor states.

Acknowledgements

We thank Glenn Langford for fruitful discussions about his work, and Aris Anagnostopoulos and Sebastiàn Urrutia for providing us their solutions. We are also grateful to Michael Trick for maintaining the TTP web page. Finally, we thank an anonymous referee for helpful comments that allowed us to improve the paper.

References

Ahuja, R. K., J. B. Orin, and D. Sharma (2000). "Very large scale neighborhood search." *International Transactions in Operations Research* 7, 301–317.

Anagnostopoulos, A., L. Michel, P. Van Hentenryck, and Y. Vergados (2005). "A Simulated Annealing Approach to the Traveling Tournament Problem." *Journal of Scheduling* (to appear), available from `http://www.cs.brown.edu/~aris/pubs/ttp.pdf`.

Araùjo, A., V. Rebello, C. Ribeiro, and S. Urrutia (2005). "A grid implementation of a GRASP-ILS heuristic for the mirrored traveling tournament problem (extended abstract)." *Proceedings of the 6th Metaheuristics International Conference (MIC2005)*, Vienna, Austria, pp. 70–76.

Birattari, M. (2004). *The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective*. Ph.D. thesis, Université Libre de Bruxelles, Brussels, Belgium.

Birattari, M. (2005). *The RACE pacakge*. URL `http://cran.r-project.org/doc/packages/race.pdf`.

Conover, W. (1999). *Practical Nonparametric Statistics*. New York, NY, USA: John Wiley & Sons, third edition.

de Werra, D. (1981). "Scheduling in Sports." P. Hansen (ed.), *Studies on Graphs and Discrete Programming*, Amsterdam, the Netherlands: North Holland, pp. 381–395.

Di Gaspero, L. and A. Schaerf (2003). "EasyLocal++: An object-oriented framework for flexible design of local search algorithms." *Software — Practice & Experience* 33(8), 733–765.

Dinitz, J. H., D. K. Garnick, and B. D. McKay (1994). "There are 526,915,620 Nonisomorphic One-factorizations of K_{12}." *Journal of Combinatorial Design* 2, 273–285.

Easton, K., G. Nemhauser, and M. Trick (2001). "The traveling tournament problem description and benchmarks." *Proceedings of the 7th International Conference on the Principles and Practice of Constraint Programming (CP-99)*, Berlin, Germany: Springer-Verlag, volume 2239 of *Lecture Notes in Computer Science*, pp. 580–589.

Easton, K., G. Nemhauser, and M. Trick (2003). "Solving the Traveling Tournament Problem: A combined integer programming and constraint programming approach." *Practice and Theory of Automated Timetabling IV (PATAT-2002)*, Berlin, Germany: Springer-Verlag, volume 2740 of *Lecture Notes in Computer Science*, pp. 100–109.

Glover, F. and M. Laguna (1997). *Tabu search*. Norwell, MA, USA: Kluwer Academic Publishers.

Hansen, P. and N. Mladenović (1999). "An Introduction to Variable Neighbourhood Search." S. Voß, S. Martello, I. Osman, and C. Roucairol (eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Norwell, MA, USA: Kluwer Academic Publishers, pp. 433–458.

Henz, M. (2001). "Scheduling a Major College Basketball Conference – Revisited." *Operations Research* 49(1), 163–168.

Jünger, M., O. Reinelt, and G. Rinaldi (1995). "The traveling salesman problem." M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (eds.), *Handbooks in Operations Research and Management Science*, Amsterdam, the Netherlands: North-Holland, chapter 7, pp. 225–330.

Langford, G. (2005). "Personal communication."

Lim, A., B. Rodrigues, and X. Zhang (2005). "A simulated annealing and hill-climbing algorithm for the traveling tournament problem." *European Journal of Operations Research* (to appear).

Lindner, C. C., E. Mendelsohn, and A. Rosa (1976). "On the Number of 1-Factorizations of the Complete Graph." *Journal of Combinatorial Theory Series B* 20, 265–282.

Lourenço, H. R., O. Martin, and T. Stützle (2002). "Applying Iterated local search to the permutation flow shop problem." F. Glover and G. Kochenberger (eds.), *Handbook of Metaheuristics*, Norwell, MA, USA: Kluwer Academic Publishers, pp. 321–353.

Mendelsohn, E. and A. Rosa (1985). "One-Factorizations of the Complete Graph – a Survey." *Journal of Graph Theory* 9, 43–65.

R Development Core Team (2005). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`, ISBN 3-900051-07-0.

Rasmussen, R. and M. Trick (2005). "A Benders approach for the constrained minimum break problems." *European Journal of Operations Research* To appear.

Ribeiro, C. C. and S. Urrutia (2004). "Heuristics for the mirrored traveling tournament problem." E. Burke and M. A. Trick (eds.), *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT-2004)*, pp. 323–342.

Schaerf, A. (1999). "Scheduling Sport Tournaments using Constraint Logic Programming." *CONSTRAINTS* 4(1), 43–65.

Trick, M. (2005). "Challenge Traveling Tournament Instances, web page." URL: `http://mat.gsia.cmu.edu/TOURN/`. Viewed: November 25, 2005, Updated: October 13, 2005.

Urrutia, S. and C. C. Ribeiro (2005). "Mazimizing Breaks and Bounding Solutions to the Mirrored Traveling Tournament Problem." *Discrete Applied Mathematics* To appear.

Wallis, W. D., A. P. Street, and J. S. Wallis (1972). *Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices*. Number 292 in Lecture Notes in Mathematics, Berlin, Germany: Springer-Verlag.