# Traveling Tournament Scheduling: A Systematic Evaluation of Simulated Annealling

Pascal Van Hentenryck and Yannis Vergados

Computer Science Department, Brown University,
Providence, RI 02912, USA
pvh@cs.brown.edu, vi@cs.brown.edu

**Abstract.** This paper considers all the variants of the traveling tournament problem (TTP) proposed in [17, 7] to abstract the salient features of major league baseball (MLB) in the United States. The variants include different distance metrics and both mirrored and non-mirrored schedules. The paper shows that, with appropriate enhancements, simulated annealing is robust across the distance metrics and mirroring. In particular, the algorithm matches or improves most best-known solutions and produces numerous new best solutions spread over all classes of problems. The main technical contribution underlying these results is a number of compositive neighborhood moves that aggregate sequences of existing moves; these novel moves preserve the mirroring or distance structure of the candidate schedule, while performing interesting transformations.

## 1 Introduction

Sport league scheduling [7] has become an important class of combinatorial optimization applications as it represents significant sources of revenue for television networks and generates extremely challenging optimization problems. In 2001, Easton, Nemhauser, and Trick [7] proposed the traveling tournament problem (TTP) to abstract the salient features of Major League Baseball (MLB) in the United States. The key to the MLB schedule is a conflict between minimizing travel distances and feasibility constraints on the home/away patterns. Travel distances are a major issue in MLB because of the number of teams and the fact that teams go on "road trips" to visit several opponents before returning home. The feasibility constraints in the MLB restricts the number of successive games that can be played at home or away. The TTP is an abstraction of the MLB intended to stimulate research in sport scheduling. A solution to the TTP is a double round-robin tournament which satisfies sophisticated feasibility constraints (e.g., no more than three away games in a row) and minimizes the total travel distances of the teams. While both minimizing the total distance traveled, and satisfying the feasibility constraints, are easy problems when considered in isolation, the combination of the two (which is captured by the TTP) makes the problem very difficult. Moreover, [7] argues that, without an approach to the TTP, it is unlikely that suitable schedules can be obtained for the MLB.

The TTP has raised significant interest in recent years since the challenge instances were proposed. The work in [7] describes both constraint and integer programming

J.C. Beck and B.M. Smith (Eds.): CPAIOR 2006, LNCS 3990, pp. 228–243, 2006.

approaches to the TTP which generate high-quality solutions. The combination of Lagrangian relaxation and constraint programming proposed in [2] improves some of the results. In 2003, we proposed an simulated algorithm exploring a large neighborhood that produced most of the best-known solutions to the instances. Our neighborhood, or a subset of it, was reused in subsequent work (e.g., [8, 12, 16]) within tabu-search and GRASP approaches or, even, simulated annealing. Recently, Rasmussen and Trick also considered Benders decomposition approaches to the TTP [15].

On the original TTP instances, our original simulated algorithm has remained most effective in producing best-known solutions, but it was never applied to the variants proposed subsequently. However, these variants may fundamentally alter the combinatorial structure of the problem. For instance, with constant distances, the order of the games in a sequence of away games is irrelevant, which is obviously not the case with the original distances. Similarly, mirroring requires that the second half of the schedule be similar to the first half but with the home-away patterns of the games reversed. Mirroring imposes severe feasibility constraints on the schedule, making it harder to find feasible tournaments and to remain in the feasible region. As a result, it was not at all clear that the algorithm would scale and perform effectively on the entire spectrum of TTP instances.

This paper originated as an attempt to determine the effectiveness and limitations of our algorithm across all TTP instances. From a practical standpoint, its main contribution is to show that the original algorithm can be enhanced to be effective across all distance metrics and mirroring. More precisely, the enhanced algorithm matches or improves most of the best known solutions for all variants and it also produces numerous new best solutions for many of the variants. From a technical standpoint, the research led to new insights into the nature of the TTP and to the following contributions:

- It shows that the algorithm can smoothly handle mirroring contraints as soft constraints by including new neighborhood moves that preserve the mirroring structure of the candidate tournament;
- It shows that the algorithm can successfully accommodate all distance metrics by including new neighborhood moves that preserve the distance structure of the candidate tournament;
- It show how to refine the original strategic oscillation scheme to the instances where feasibility constraints are much stronger.

The novel neighborhood moves are in fact sequences of existing moves. As such, they do not improve the connectivity of the neighborhood for the TTP. Their significance comes from the fact that, in the original algorithm, these sequences have a low probability of taking place, although they capture fundamental aspects of the problem structure.

The rest of the paper is organized as follows. Section 2 describes the various travelling tournament problems. Section 3 reviews the original simulated sinnealing algorithm. Section 4 explains how to adapt the algorithm to accommodate mirroring, Section 5 presents new neighborhood moves for exploiting the distance structure, and Section 6 presents the remaining algorithmic enhancements. Section 7 presents the experimental results and Section 8 concludes the paper.

## 2 Problem Description

The traveling tournament problem, as originally introduced in [7], was first studied only for benchmarks based on the distances between cities of the 16 teams of National League Baseball. The solution space consisted of all double round-robin schedules and did not include mirroring. In recent years, increased attention to the problem has led to the study of a variety of variants. On one hand, the problem was studied under the additional constraint that the schedule be a *mirrored* double round-robin tournament. On the other hand, alternative sets of distances between cities were considered. Furthermore, the problem was been studied for larger number of teams (up to 24). This section describes the TTP and its various variants.

**The Basic TTP Problem.** A TTP input consists of $n$ teams ($n$ even) and an $n \times n$ symmetric matrix $d$, such that $d_{ij}$ represents the distance between the homes of teams $T_i$ and $T_j$. A solution is a schedule in which each team plays with each other twice, once in each team's home. Such a schedule is called a *double round-robin tournament*. It should be clear that a double round-robin tournament has $2n - 2$ rounds. For a given schedule $S$, the cost of a team as the total distance that it has to travel starting from its home, playing the scheduled games in $S$, and returning back home. The cost of a solution is defined as the sum of the cost of every team. The goal is to find a schedule with minimum cost satisfying the following two constraints:

1. **Atmost Constraints:** No more than three consecutive home or away games are allowed for any team.
2. **Norepeat Constraints:** A game of $T_i$ at $T_j$'s home cannot be followed by a game of $T_j$ at $T_i$'s home.

As a consequence, a double round-robin schedule is feasible if it satisfies the atmost and norepeat constraints and is infeasible otherwise.

In this paper, a schedule is represented by a table indicating the opponents of the teams. Each line corresponds to a team and each column corresponds to a round. The opponent of team $T_i$ at round $r_k$ is given by the absolute value of element $(i, k)$. If $(i, k)$ is positive, the game takes place at $T_i$'s home, otherwise at $T_i$'s opponent home. Consider for instance the schedule $S$ for 6 teams (and thus 10 rounds).

| T\R | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | 3 | -5 | -4 | -3 | 5 | 2 | -6 |
| 2 | 5 | 1 | -3 | -6 | 4 | 3 | 6 | -4 | -1 | -5 |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | -6 | -5 | 4 |
| 4 | 3 | 6 | -1 | -5 | -2 | 1 | 5 | 2 | -6 | -3 |
| 5 | -2 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 3 | 4 | 1 |

Schedule $S$ specifies that team $T_1$ has the following schedule. It successively plays against teams $T_6$ at home, $T_2$ away, $T_4$ at home, $T_3$ at home, $T_5$ away, $T_4$ away, $T_3$ away, $T_5$ at home, $T_2$ at home, $T_6$ away. The travel cost of team $T_1$ is

$$d_{12} + d_{21} + d_{15} + d_{54} + d_{43} + d_{31} + d_{16} + d_{61}.$$

Observe that long stretches of games at home do not contribute to the travel cost but are limited by the atmost constraints.

**Mirroring.** In some sports leagues (e.g., in most European soccer leagues), it is common practice to adopt a two-stage mirrored schedule. If $n$ is the number of participating teams, each stage has $n-1$ rounds and the $n-1$ rounds of the second stage are simply a copy of the the first-stage rounds with a swap of home/away patterns of each game. In terms of the table representation, a schedule is *mirrored* if

$$\forall i, j:\ 1 \leq j \leq n-1:\ S[i, j+n-1] = -S[i, j].$$

**Distance Metrics.** All the distance metrics studied in this paper can be found on the web site [17]. They are defined as follows:

- [NL$n$:] For $n = 4, 6, \ldots, 16$, NL$n$ is the distances between the subset of the first $n$ teams of National League Baseball.
- [Circular:] The $n$ teams are labeled with numbers 0 through $n-1$ and are placed on a circle in this order. Then, for any two teams, the distance is given by the length of the shortest arc connecting the teams. These distances satisfy

$$\forall i > j:\ d_{ij} = \min\{i-j, j-i+n\}.$$

- [Constant:] The $n$ teams are at unit distance to one another, i.e., $d_{ij} = 1$, for all $i, j$.

For every distance metric, we are interested in solutions to both the mirrored and the non-mirrored version of the TTP.

## 3 The Original Simulated Annealing Algorithm

Our simulated algorithm TTSA in [1] is based on four main design decisions:

1. Constraints are separated into two groups: hard constraints, which are always satisfied by the configurations, and soft constraints, which may or may not be violated. The hard constraints are the round-robin constraints, while the soft constraints are the norepeat and atmost constraints. In other words, all configurations in the search represents a double round-robin tournament, which may or may not violate the norepeat and atmost constraints. To drive the search toward feasible solutions, TTSA modifies the original objective function to include a penalty term.
2. TTSA is based on a large neighborhood of size $O(n^3)$, where $n$ is the number of teams. In addition, these moves may affect significant portions of the configurations. For instance, they may swap the schedule of two teams, which affects $4(n-2)$ entries in a configuration. In addition, some of these moves can be regarded as a form of ejection chains sometimes used in tabu search [11, 14].

3. TTSA dynamically adjusts the objective function to balance the time spent in the feasible and infeasible regions. This adjustment resembles the strategic oscillation idea [9] successfully in tabu search to solve generalized assignment problems [5], although the details differ since simulated annealing is used as the metaheuristics.
4. TTSA also uses reheats (e.g., [3]) to escape local minima at low temperatures. The "reheats" increase the temperature again and divide the search in several phases.

The rest of this section explore some of these aspects in more detail, as they are pertinent to the understanding of this paper. Since they are double round-robin tournaments, configurations are called schedules in the following. Observe that, contrary to simpler problems such as break minimization [18], we found it critical to explore the infeasible region in the TPP and to consider moves that significantly alter the schedules. In our experiments, neighborhoods consisting of simpler moves or considering only feasible schedules tend to produce local minima of low quality. Of course, this does not mean that such neighborhoods do not exist, simply that we have not be able to design them so far. This difference between the TTP and the break minimization stems from the fact that break minimization assumes a fixed schedule: only the home/away patterns must be determined. In contrast, the objective function in the TTP must not only determine the home/away patterns; it must also determine the schedule of each team to minimize the total travel distance. This additional difficulty and the tension it produces with the feasibility constraints is what makes the TTP particularly challenging.

### 3.1 The Neighborhood

The neighborhood of a schedule $S$ is the set of the (possibly infeasible) schedules which can be obtained by applying one of five types of moves. The first three types of moves have a simple intuitive meaning, while the last two generalize them.

***SwapHomes***$(S, T_i, T_j)$**.** This move swaps the home/away roles of teams $T_i$ and $T_j$. In other words, if team $T_i$ plays home against team $T_j$ at round $r_k$, and away against $T_j$'s home at round $l$, *SwapHomes*$(S, T_i, T_j)$ is the same schedule as $S$, except that now team $T_i$ plays away against team $T_j$ at round $r_k$, and home against $T_j$ at round $r_l$. There are $O(n^2)$ such moves. Consider the schedule $S$:

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | 3 | -5 | -4 | -3 | 5 | 2 | -6 |
| **2** | 5 | 1 | -3 | -6 | 4 | ■ | 6 | -4 | ■ | -5 |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | -6 | -5 | 4 |
| **4** | 3 | 6 | -1 | -5 | -2 | ■ | 5 | 2 | ■ | -3 |
| 5 | -2 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 3 | 4 | 1 |

The move *SwapHomes*$(S, T_2, T_4)$ produces the schedule:

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | 3 | -5 | -4 | -3 | 5 | 2 | -6 |
| **2** | 5 | 1 | -3 | -6 | -4 | | 6 | 4 | | -5 |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | -6 | -5 | 4 |
| **4** | 3 | 6 | -1 | -5 | 2 | | 5 | -2 | | -3 |
| 5 | -2 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 3 | 4 | 1 |

***SwapRounds*$(S, r_k, r_l)$.** The move simply swaps rounds $r_k$ and $r_l$. There are also $O(n^2)$ such moves. Consider the schedule $S$:

| **T\R** | 1 | 2 | **3** | 4 | **5** | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | | -5 | | -3 | 5 | 2 | -6 |
| 2 | 5 | 1 | -3 | | 4 | | 6 | -4 | -1 | -5 |
| 3 | -4 | 5 | 2 | | 6 | | 1 | -6 | -5 | 4 |
| 4 | 3 | 6 | -1 | | -2 | | 5 | 2 | -6 | -3 |
| 5 | -2 | -3 | 6 | | 1 | | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -5 | | -3 | | -2 | 3 | 4 | 1 |

The move *SwapRounds*$(S, r_3, r_5)$ produces the schedule

| **T\R** | 1 | 2 | **3** | 4 | **5** | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | -5 | | 4 | | -3 | 5 | 2 | -6 |
| 2 | 5 | 1 | 4 | | -3 | | 6 | -4 | -1 | -5 |
| 3 | -4 | 5 | 6 | | 2 | | 1 | -6 | -5 | 4 |
| 4 | 3 | 6 | -2 | | -1 | | 5 | 2 | -6 | -3 |
| 5 | -2 | -3 | 1 | | 6 | | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -3 | | -5 | | -2 | 3 | 4 | 1 |

***SwapTeams*$(S, T_i, T_j)$.** This move swaps the schedule of Teams $T_i$ and $T_j$ (except, of course, when they play against each other). There are $O(n^2)$ such moves again. Consider the schedule $S$:

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | 3 | -5 | -4 | -3 | 5 | 2 | -6 |
| **2** | 5 | 1 | -3 | -6 | 4 | 3 | 6 | -4 | -1 | |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | -6 | -5 | 4 |
| 4 | 3 | 6 | -1 | -5 | -2 | 1 | 5 | 2 | -6 | -3 |
| **5** | -2 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 3 | 4 | 1 |

The move *SwapTeams*$(S, T_2, T_5)$ produces the schedule

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -5 | ■ | 3 | -2 | ■ | -3 | 2 | 5 | ■ |
| **2** | 5 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | ■ |
| 3 | -4 | 2 | 5 | ■ | 6 | -5 | ■ | -6 | -2 | ■ |
| 4 | 3 | 6 | -1 | -2 | -5 | ■ | 2 | 5 | ■ | -3 |
| **5** | -2 | 1 | -3 | -6 | 4 | 3 | 6 | -4 | -1 | ■ |
| 6 | -1 | -4 | -2 | 5 | ■ | 2 | -5 | ■ | 4 | 1 |

Note that, in addition to the changes in lines 2 and 5, the corresponding lines of the opponents of $T_i$ and $T_j$ must be changed as well. As a consequence, there are four values per round (column) that are changed (except when $T_i$ and $T_j$ meet).

It turns out that these three moves are not sufficient for exploring the entire search space and, as a consequence, they lead to suboptimal solutions for large instances. To improve these results, it is important to consider two, more general, moves. Although these moves do not have the apparent interpretation of the first three, they are similar in structure and they significantly enlarge the neighborhood, resulting to a more connected search space. More precisely, these moves are partial swaps: they swap a subset of the schedule in rounds $r_i$ and $r_j$ or a subset of the schedule for teams $T_i$ and $T_j$. The benefits from these moves come from the fact that they are not as global as the "macro"-moves *SwapTeams* and *SwapRounds*. As a consequence, they may achieve a better tradeoff between feasibility and optimality by improving feasibility in one part of the schedule, while not breaking feasibility in another one. They are also more "global" than the "micro"-moves *SwapHomes*.

***PartialSwapRounds*$(S, T_i, r_k, r_l)$.** This move considers team $T_i$ and swaps its games at rounds $r_k$ and $r_l$. The rest of the schedule for rounds $r_k$ and $r_l$ is updated to produce a double round-robin tournament. Consider the schedule $S$

| **T\R** | 1 | **2** | 3 | 4 | 5 | 6 | 7 | 8 | **9** | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 2 | 3 | -5 | -4 | -3 | 5 | 4 | -6 |
| **2** | 5 | 1 | ■ | -5 | 4 | 3 | 6 | -4 | -6 | ■ |
| 3 | -4 | 5 | 4 | -1 | 6 | -2 | 1 | -6 | -5 | 2 |
| 4 | 3 | 6 | -3 | -6 | -2 | 1 | 5 | 2 | -1 | -5 |
| 5 | -2 | -3 | 6 | 2 | 1 | -6 | -4 | -1 | 3 | 4 |
| 6 | -1 | -4 | -5 | 4 | -3 | 5 | -2 | 3 | 2 | 1 |

and the move *PartialSwapRounds*$(S, T_2, r_2, r_9)$. Obviously swapping the game in rounds $r_2$ and $r_9$ would not lead to a round-robin tournament. It is also necessary to swap the games of team 1, 4, and 6 in order to obtain:

| **T\R** | 1 | **2** | 3 | 4 | 5 | 6 | 7 | 8 | **9** | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | 4 | | 3 | -5 | -4 | -3 | 5 | -2 | |
| **2** | 5 | -6 | | -5 | 4 | 3 | 6 | -4 | 1 | |
| 3 | -4 | 5 | 4 | -1 | 6 | -2 | 1 | -6 | -5 | 2 |
| 4 | 3 | -1 | | -6 | -2 | 1 | 5 | 2 | 6 | |
| 5 | -2 | -3 | 6 | 2 | 1 | -6 | -4 | -1 | 3 | 4 |
| 6 | -1 | 2 | | 4 | -3 | 5 | -2 | 3 | -4 | |

This move, and the next one, can thus be regarded as a form of ejection chain [11, 14].

Finding which games to swap is not difficult: it suffices to find the connected component which contains the games of $T_i$ in rounds $r_k$ and $r_l$ in the graph where the vertices are the teams and where an edge contains two teams if they play against each other in rounds $r_k$ and $r_l$. All the teams in this component must have their games swapped. Note that there are $O(n^3)$ such moves.

***PartialSwapTeams***$(S, T_i, T_j, r_k)$**.** This move considers round $r_k$ and swaps the games of teams $T_i$ and $T_j$. Then, the rest of the schedule for teams $T_i$ and $T_j$ (and their opponents) is updated to produce a double round-robin tournament. Note that, as was the case with *SwapTeams*, four entries per considered round are affected. There are also $O(n^3)$ such moves. Consider the schedule $S$

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **9** | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 4 | 3 | -5 | -4 | -3 | 5 | 2 | -6 |
| **2** | 5 | 1 | -3 | -6 | 4 | 3 | 6 | -4 | -1 | |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | -6 | -5 | 4 |
| **4** | 3 | 6 | -1 | -5 | -2 | 1 | 5 | 2 | -6 | |
| 5 | -2 | -3 | 6 | 4 | 1 | -6 | -4 | -1 | 3 | 2 |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 3 | 4 | 1 |

The move *PartialSwapRounds*$(S, T_2, T_4, r_9)$ produces the schedule

| **T\R** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **9** | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | -2 | 2 | | -5 | -4 | -3 | 5 | 4 | |
| **2** | 5 | 1 | -1 | -5 | | 3 | 6 | -4 | -6 | -3 |
| 3 | -4 | 5 | 4 | | 6 | -2 | 1 | -6 | -5 | 2 |
| **4** | 3 | 6 | -3 | -6 | | 1 | 5 | 2 | -1 | -5 |
| 5 | -2 | -3 | 6 | 2 | | -6 | -4 | -1 | 3 | 4 |
| 6 | -1 | -4 | -5 | 4 | | 5 | -2 | 3 | 2 | |

### 3.2 The Objective Function

As mentioned already, the configurations in algorithm TTSA are schedules which may or may not satisfy the norepeat and atmost constraints. To drive toward feasible

solution, the standard objective function function $cost$ is replaced by a more complex objective function which combines travel distances and the number of violations. The new objective function $C$ is defined as follows:

$$C(S) = \begin{cases} cost(S) \text{ if } S \text{ is feasible,} \\ \sqrt{cost(S)^2 + [w \cdot f(nbv(S))]^2} \text{otherwise,} \end{cases}$$

where $nbv(S)$ denotes the number of violations of the norepeat and atmost constraints, $w$ is a weight, and $f : N \rightarrow N$ is a sublinear function such that $f(1) = 1$.

It is interesting to give the rationale behind the choice of $f$. The intuition is that the first violation costs more than subsequent ones, since adding 1 violation to a schedule with 6 existing ones does not make much difference. More precisely, crossing the feasible/infeasible boundary costs $w$, while $v$ violations only cost $wf(v)$, where $f(v)$ is sublinear in $v$. In our experiments, we chose $f(v) = 1 + (\sqrt{v} \ln v)/\lambda$ with $\lambda$ equal to 2 on small instances and 1 on larger ones. This choice makes sure that $f$ does not grow too slowly to avoid solutions with very many violations.

### 3.3 Strategic Oscillation

TTSA also includes a strategic oscillation strategy often used in tabu search when the search explores both the feasible and infeasible regions (e.g., [9, 5]). The key idea is to vary the weight parameter $w$ during the search. In advanced tabu-search applications (e.g., [5]), the penalty is updated according to the frequencies of feasible and infeasible configurations in the last iterations. Such a strategy is meaningful in that context, but is not particularly appropriate for simulated annealing since very few moves may be selected. TTSA uses a very simple scheme. Each time it generates a new best solution TTSA multiplies $w$ by some constant $\delta > 1$ if the new solution is infeasible or divide $w$ by some constant $\theta > 1$ if the new solution is feasible.

The rationale here is to keep a balance between the time spent exploring the feasible region and the time spent exploring infeasible schedule. After having spent a long time in the infeasible region, the weight $w$, and thus the penalty for violations, will become large and it will drive the search toward feasible solutions. Similarly, after having spent a long time in the feasible region, the weight $w$, and thus the penalty for violations, will become small and it will drive the search toward infeasible solutions. In our experiments, we chose $\delta = \theta$ for simplicity.

## 4 Mirroring

This section reviews the enhancements of the algorithm to find mirrored tournaments.

### 4.1 Mirroring Constraints

Mirrored constraints, like atmost and norepeat constraints, are considered soft constraints in the algorithm, since restricting the neighborhood graph to only mirrored schedules was not found effective. In other words, the neighborhood graph consists

of nodes representing both mirrored and non-mirrored schedules and it is the role of the objective function to drive the search toward mirrored schedules. More precisely, the algorithm associates a soft mirrored constraint with each entry $S[i,j]$ $(1 \leq i \leq n \;\&\; 1 \leq j < n-1)$ and the constraint holds when

$$S[i,j] = -S[i,j+n-1]$$

It is thus possible to include the violations of these constraints in the objective function as was the case for the atmost and norepeat constraints.

Unfortunately, this simple modeling is not directly effective given the large number of mirroring constraints that can be violated in candidate schedules. As a result, the algorithm weights the mirroring constraints appropriately and rewrite the function $nbv(S)$ of the objective function as the sum

$$nbv_a(S) + nbv_r(S) + \frac{nbv_m(S)}{\mu}$$

where $nbv_a$, $nbv_r$, and $nbv_m$ represent the violations of the atmost, norepeat, and mirroring constraints respectively. In particular, the violations of the mirroring constraints is simply defined as

$$nbv_m(S) = |\{(i,j) : S[i,j] \neq -S[i,j+n-1] \;\&\; 1 \leq i \leq n \;\&\; 1 \leq j \leq n-1\}|.$$

### 4.2 Mirroring Moves

Once the algorithm reaches a (possibly infeasible) mirrored schedule, it is beneficial to let the search explored neighboring *mirrored* schedules with fewer violations of the remaining constraints or smaller distances. The simulated-annealing algorithm, with its present moves, has a low probability of exploring such moves. It is thus important to design mirrored versions of the moves that affect the mirroring constraints: the *SwapRounds*, *PartialSwapRounds*, and *PartialSwapTeams* moves. The basic idea behind the aggregate moves is to apply the original moves to both parts of the schedule simultaneously. For instance, the new move $SwapRoundsMirrored(S, r_k, r_l)$ for $1 \leq r_k < r_l \leq n-1$ is the aggregate

$$(SwapRounds(S, r_k, r_l), SwapRounds(S, T_i, r_k + n - 1, r_l + n - 1)).$$

Mirroring constraints are invariant with respect to mirrored moves: in particular, if the schedule is mirrored, it remains so. As a result, the algorithm is able to preserve the structure of the schedule with respect to mirroring constraints, while performing transformations that affect the remaining constraints or the distances.

## 5 Distance Metrics

This section presents two additional composite moves that affect the distances in interesting ways. Once again, the novel moves aggregate sequences of existing moves that have a low probability of taking place in the original algorithm. Hence, they also

preserve some significant structure in the schedule, while performing some interesting transformations.

The key idea underlying the novel moves is to reverse subsequences of away moves. Recall that travel only occurs for successive pairs of away games and for successive pairs of (home,away) and (away,home) games. So, by reversing a subsequence of away games, we preserve a significant part of the distance structure, while modifying it in a way that is difficult to achieve by sequences of original moves. In particular, the distances in the reversed subsequence, as well as the distances in the sequence of the other teams that must also be reversed to maintain a double round-robin tournament, remain the same. It is only at the beginning and at the end of the subsequences that distances are changing. In fact, moves similar in spirit are also used in car sequencing [4] but they are simpler since they do not have to account for the round-robin constraints and the distance structure.

The algorithm thus considers moves of the form *ReverseAwayRun*$(S, T_i, r_k, m)$ where team $T_i$ plays an away sequence from round $r_k$ to round $r_{k+m}$. The effect of the move is similar to the sequence of $p = (m+1)/2$ moves

*PartialSwapRounds*$(S, T_i, r_k, r_{k+m})$
*PartialSwapRounds*$(S, T_i, r_{k+1}, r_{k+m-1})$
...
*PartialSwapRounds*$(S, T_i, r_{k+p-1}, r_{k+m+1-p})$

For instance, consider the schedule $S$

| **T\R** | 1 | 2 | 3 | 4 | 5 | **6** | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 6 | 2 | 4 | 3 | -5 | -4 | -3 | -6 | -2 | |
| 2 | 5 | -1 | -3 | -6 | 4 | 3 | 6 | -5 | 1 | -4 |
| 3 | -4 | 5 | 2 | -1 | 6 | -2 | 1 | 4 | -5 | -6 |
| 4 | 3 | 6 | -1 | -5 | -2 | 1 | 5 | -3 | -6 | 2 |
| 5 | -2 | -3 | 6 | 4 | 1 | -6 | -4 | 2 | 3 | -1 |
| 6 | -1 | -4 | -5 | 2 | -3 | 5 | -2 | 1 | 4 | 3 |

The move *ReverseAwayRun*$(S, T_1, r_6, 4)$ produces the schedule $S'$:

| **T\R** | 1 | 2 | 3 | 4 | 5 | **6** | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 6 | 2 | 4 | 3 | -5 | -2 | -6 | -3 | -4 | |
| 2 | 5 | -1 | -3 | -6 | 4 | 1 | -5 | 6 | 3 | |
| 3 | -4 | 5 | 2 | -1 | 6 | -5 | 4 | 1 | -2 | |
| 4 | 3 | 6 | -1 | -5 | -2 | -6 | -3 | 5 | 1 | |
| 5 | -2 | -3 | 6 | 4 | 1 | 3 | 2 | -4 | -6 | |
| 6 | -1 | -4 | -5 | 2 | -3 | 4 | 1 | -2 | 5 | |

When $d_{52} + d_{41} < d_{54} + d_{21}$, the move improves the distance with respect to team $T_1$, without affecting the atmost violations of team $T_1$ and the distance structure inside the

subsequence. There are several points worth highlighting here. First, reversing entire sequences of away games does not change the distance for the team considered and should not be considered. Second, the value $m$ is never very large in the TTP instances, since the atmost constraints drive the search toward small subsequences of away games. Finally, the algorithm must include a mirrored version *ReverseAwayRunMirrored* of the moves since they affect the mirroring constraints.

## 6 Algorithmic Refinements

**Strategic Oscillation.** The mirroring constraints make it harder to find feasible tournaments and the search may spend considerable time in the infeasible region before finding a first feasible solution. As a result, even small values for $\mu$ and $\lambda$, the strategic oscillation scheme will overly inflate the violation weight $w$, leading the search to stagnation. To alleviate this pathological case, the algorithm now takes a two-step approach. In a first phase, which lasts until the first feasible tournament is found, no oscillation takes place. In the second phase, the strategic oscillation scheme is activated as before and balances the time spent in the infeasible and feasible regions. Note also the synergy between this scheme and the new neighborhood moves. By including mirrored moves, the algorithm is able to better balance the time it spends in the feasible and infeasible regions in presence of mirroring constraints.

**Initial Schedules.** The simple backtrack search used in [1] to find initial schedules does not scale well when the number of teams increases, which is the case in the constant and circular variants. As a result, like in [16], the algorithm now uses a randomized version of the hill-climbing algorithm for generating 1-factorizations [6]. The initial schedules generated by this randomized algorithm are more diversified than perturbations of schedules obtained by the polygon algorithm. The algorithm in [6] works for single round-robin schedules but a double round-robin schedule can be obtained by a simple mirroring.

## 7 Experimental Results

The enhanced version of the algorithm was run on all the mirrored and non-mirrored instances given on Michael Trick's webpage [17].[1] For each instance, 20 experiments were carried out from randomly chosen schedules on an AMD Athlon 64 at 2Ghz. The results are reported in two tables for each variant. The first table reports the best, mean, and worst solutions found by the algorithm, as well as the standard deviation and the best known solution value at the time of writing. The second table reports the time to reach the best solution, the mean time of each experiment, and the standard deviation. Bold face indicates improved results. It is also important to mention that many authors (e.g., [8, 12, 16]) now use the neighborhood we originally proposed in [1] which makes it much harder to improve the results (since, in a sense, we are also competing with ourselves). Our implementation is also slightly more incremental than in 2003, but this is not seen as a major factor in these results in contrast to the new moves proposed herein.

[1] They do not include the NFL instances just posted in December.

**Table 1.** Solution Quality and Solution Times for the NL$n$ Distances with Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 41928 | 41928 | 43025 | 42037.65 | 291.98 |
| 10 | 63832 | 63832 | 64409 | 63860.85 | 125.75 |
| 12 | 120665 | **119608** | 120774 | **120121.55** | 417.07 |
| 14 | 208086 | **199363** | 210599 | **202400.50** | 2883.39 |
| 16 | 279618 | **279077** | 297173 | 284036.95 | 4770.61 |

| $n$ | T for min | mean(T) | std(T) |
|---|---|---|---|
| 8 | 0.1 | 1555.55 | 1880.94 |
| 10 | 477.2 | 8511.29 | 17132.49 |
| 12 | 15428.1 | 49373.31 | 32834.88 |
| 14 | 34152.3 | 70898.90 | 48551.27 |
| 16 | 55640.8 | 47922.16 | 36948.40 |

**Table 2.** Solution Quality and Solution Times for the Constant Distances with Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 80 | 80 | 80 | 80 | 0 |
| 10 | 130 | 130 | 130 | 130 | 0 |
| 12 | 192 | 192 | 192 | 192 | 0 |
| 14 | 253 | 253 | 253 | 253 | 0 |
| 16 | 342 | 342 | 342 | 342 | 0 |
| 18 | 432 | 432 | 432 | 432 | 0 |
| 20 | 524 | **522** | **522** | **522** | 0 |
| 22 | 650 | 650 | 650 | 650 | 0 |
| 24 | 768 | 768 | 768 | 768 | 0 |

| $n$ | T for min | mean(T) | std(T) |
|---|---|---|---|
| 8 | 0.1 | 0.06 | 0 |
| 10 | 0.1 | 0.10 | 0 |
| 12 | 0.3 | 0.56 | 0.38 |
| 14 | 6.0 | 154.26 | 147.95 |
| 16 | 2.7 | 3.29 | 1.53 |
| 18 | 8.1 | 24.60 | 19.20 |
| 20 | 1106.3 | 12556.20 | 10347.58 |
| 22 | 24.3 | 45.42 | 22.90 |
| 24 | 813.3 | 1791.77 | 983.47 |

**Table 3.** Solution Quality and Solution Times for the Circular Distances with Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 140 | 140 | 140 | 140 | 0 |
| 10 | 272 | 272 | 276 | 273.60 | 1.01 |
| 12 | 456 | **432** | **444** | **434.90** | 3.12 |
| 14 | 714 | **696** | 726 | **708.90** | 7.05 |
| 16 | 978 | **968** | 1072 | 1001.60 | 28.55 |
| 18 | 1306 | 1352 | 1364 | 1357.80 | 3.40 |
| 20 | 1882 | **1852** | 2198 | 2017.60 | 60.64 |

| $n$ | T for min | mean(T) | std(T) |
|---|---|---|---|
| 8 | 0.2 | 74.18 | 55.13 |
| 10 | 28160.0 | 12527.18 | 12208.25 |
| 12 | 93.1 | 4658.58 | 3560.27 |
| 14 | 53053.5 | 23549.14 | 16311.15 |
| 16 | 38982.7 | 23360.81 | 14451.53 |
| 18 | 178997.5 | 106139.77 | 57175.01 |
| 20 | 59097.9 | 43137.13 | 22515.46 |

**Mirrored Instances.** Tables 1, 2, and 3 report the results for mirrored instances, which are particularly impressive. The algorithm matches or improves all best-known solutions (but one). It produces 8 new best solutions and the improvements essentially occur for larger instances. This was a surprising result for us, since we thought that mirroring instances would be significantly more challenging for the algorithm. Some of the improvements may also be quite large and reach more than 4%.

**Non-mirroring Instances.** Tables 4, 5, and 6 report the results for the non-mirrored instances. On the NL$n$ and constant distance metrics, the algorithm is once again impressive, matches or improves all the best-known solutions, and improves 6 instances. Once again, the gains are obtained on the larger instances. The results on the circular

**Table 4.** Solution Quality and Solution Times for the NL$n$ Distances without Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 39721 | 39721 | 39721 | 39721 | 0 |
| 10 | 59436 | 59436 | 59583 | 59561.63 | 48.33 |
| 12 | 111483 | **111248** | 116018 | 112663.32 | 738.55 |
| 14 | 190056 | **189156** | 195742 | 193187.85 | 1432.99 |
| 16 | 270794 | **267194** | 282005 | 273552.64 | 3461.49 |

| $n$ | T for min | mean(T) | std(T) |
|---|---|---|---|
| 8 | 1169.0 | 1639.33 | 332.38 |
| 10 | 2079.6 | 27818.24 | 64873.91 |
| 12 | 202756.2 | 150328.30 | 92385.48 |
| 14 | 90861.4 | 77587.86 | 40346.49 |
| 16 | 344633.4 | 476191.65 | 389371.71 |

**Table 5.** Solution Quality and Solution Times for the Constant Distances without Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 80 | 80 | 80 | 80 | 0 |
| 10 | 124 | 124 | 124 | 124 | 0 |
| 12 | 181 | 181 | 181 | 181 | 0 |
| 14 | 252 | 252 | 252 | 252 | 0 |
| 16 | 327 | 327 | 329 | 328 | 0.31 |
| 18 | 418 | **417** | 418 | **417.65** | 0.47 |
| 20 | 521 | **520** | 522 | **520.90** | 0.53 |
| 22 | 632 | **628** | **630** | **629.40** | 0.58 |
| 24 | 757 | **750** | **753** | **750.65** | 0.91 |

| $n$ | min(T) | mean(T) | std(T) |
|---|---|---|---|
| 8 | 0.2 | 0.14 | 0.14 |
| 10 | 4.6 | 3.96 | 2.43 |
| 12 | 128.7 | 1126.85 | 1480.45 |
| 14 | 26.1 | 95.32 | 59.42 |
| 16 | 82884.1 | 16042.20 | 22332.36 |
| 18 | 10362.8 | 7091.27 | 6614.78 |
| 20 | 7781.7 | 22850.72 | 25094.76 |
| 22 | 39380.3 | 22618.46 | 20364.85 |
| 24 | 16356.6 | 28941.04 | 22987.96 |

**Table 6.** Solution Quality and Solution Times for the Circular Distances without Mirroring

| $n$ | Old Best | min(D) | max(D) | mean(D) | std(D) |
|---|---|---|---|---|---|
| 8 | 132 | 132 | 132 | 132.00 | 0 |
| 10 | 242 | 242 | 256 | 252.70 | 3.24 |
| 12 | 408 | 420 | 432 | 427.13 | 3.43 |
| 14 | 654 | 666 | 690 | 679.70 | 5.14 |
| 16 | 928 | 968 | 1072 | 1001.60 | 28.55 |
| 18 | 1306 | 1352 | 1364 | 1357.80 | 3.40 |
| 20 | 1842 | 1852 | 2198 | 2017.60 | 60.64 |

| $n$ | T for min | mean(T) | std(T) |
|---|---|---|---|
| 8 | 3.2 | 589.23 | 590.74 |
| 10 | 19261.6 | 14491.27 | 7937.21 |
| 12 | 151459.1 | 96717.13 | 52788.38 |
| 14 | 12908.9 | 86766.67 | 68408.73 |
| 16 | 38982.7 | 23360.81 | 14451.53 |
| 18 | 178997.5 | 106139.77 | 57175.01 |
| 20 | 59097.9 | 43137.13 | 22515.46 |

instances are somewhat disappointing. The algorithm cannot match the best-known results on the larger instances, although it is often very close to the best-known solutions. This may be due to the fact that the algorithm only uses mirrored starting schedules, which may bias the search. In fact, the best solutions found by our algorithm for 16 and 20 teams are mirrored schedules. These instances need to be investigated more carefully, since very little time was spent tuning the parameters.

## 8 Conclusion

This paper reconsidered our original simulated algorithm for the travelling tournament problem (TTP) and studied its effectiveness across all TTP variants. The variants in-

clude various distance metrics, as well as mirroring constraints. From a practical standpoint, its main contribution is to show that the original algorithm can be enhanced to be effective across all distance metrics and mirroring. The main technical novelty in the algorithm is the introduction of novel neighborhood moves that capture sequences of earlier moves. As such, these novel moves do not improve the connectivity of the neighborhood for the TTP. Their significance comes from the fact that, in the original algorithm, these sequences have a low probability, although they capture fundamental aspects of the mirroring or distance structure. The resulting algorithm matches or improves most best-known solutions and it also produces numerous new best solutions for many of the variants. It is thus quite remarkable that a single algorithm be so robust in producing high-quality solutions to all instances.

An important area of future research is to find high-quality solutions more quickly. For instance, Gaspero and Schaerf [8] embedded a subset of our neighborhood in a tabu-search algorithm and obtained high-quality solutions quickly, although their best solutions still do not match our best found solutions. Whether it is possible to find the same solution quality with the speed of their algorithm is an important issue to address.

## References

1. A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. A Simulated Annealing Approach to the Traveling Tournament Problem In *CP-AI-OR'2003*, Montreal, Canada, May 2003.
2. T. Benoist, F. Laburthe, and B. Rottembourg. Lagrange Relaxation and Constraint Programming Collaborative Schemes for Travelling Tournament Problems. In *CP-AI-OR'2001*, Wye College (Imperial College), Ashford, Kent UK, April 2001.
3. D.T. Connelly. General Purpose Simulated Annealing. *European Journal of Operations Research*, 43, 1992.
4. A. Davenport and E. Tsang. Solving Constraint Satisfaction Sequencing Problems by Iterative Repair. In *Procceedings of the First International Conference on the Practical Applications of Constraint Technologies and Logic Programming (PACLP-99)*, pages 345–357, London, April 1999.
5. Juan A. Díaz and Elena Fernández. A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research*, 132(1):22–38, July 2001.
6. J. H. Dinitz, and D. R. Stinson. A Hill-climbing Algorithm for the Construction of One-Factorizations and Room Squares. *SIAM J. Alg. Disc. Meth.*, 8(3):430–438, July 1987.
7. K. Easton, G. Nemhauser, and M. Trick. The traveling tournament problem description and benchmarks. In *Seventh International Conference on the Principles and Practice of Constraint Programming (CP'01)*, pages 580–589, Paphos, Cyprus, 2001. Springer-Verlag, LNCS 2239.
8. L. Di Gaspero, and A. Schaerf. A Tabu Search Approach to the Traveling Tournament Problem. In *Proceedings of RCRA 2005, Associazione Italiana per l'Intelligenza Artificiale (AI*IA)*, pages 23–27, Ferrara, Italy, June 2005
9. F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
10. S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.
11. M. Laguna, J.P. Kelly, Gonzalez-Velarde, and F. Glover. Tabu search for the multilevel generalized assignment problems. *European Journal of Operational Research*, 42:677–687, 1995.

12. A. Lim, B. Rodrigues and X. Zhang. Scheduling Sports Competitions at Multiple Venues Revisited. European Journal of Operational Research, 2005 (Accepted for Publication).
13. I. H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.
14. E. Pesch and F. Glover. TSP Ejection Chains. *Discrete Applied Mathematics*, 76:165–181, 1997.
15. R. Rasmussen and M. Trick. A Benders Approach to the Constrained Minimum Break Problem. European Journal of Operational Research, 2005 (Accepted for Publication)
16. C. C. Ribeiro, and S. Urrutia. Heuristics for the Mirrored Traveling Tournament Problem. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT'04)*, 323-342, 2004.
17. M. Trick. http://mat.gsia.cmu.edu/TOURN/, 2002.
18. P. Van Hentenryck and Y. Vergados Minimizing Breaks in Sport Scheduling with Local Search In *Proceedings of the 15th International Conference on Automated Planning and Scheduling* Monterey, CA, June 2005.