Mirrored Traveling Tournament Problem: An Evolutionary Approach

2 authors, including:

Luiz A. N. Lorena
National Institute for Space Research
229 PUBLICATIONS   **3,674** CITATIONS

Mirrored Traveling Tournament Problem: An Evolutionary Approach

Fabrício Lacerda Biajoli and Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais - INPE
Laboratório Associado de Computação e Matemática Aplicada - LAC
Av. dos Astronautas, 1.758 - São José dos Campos-SP, Brasil

Abstract. The *Mirrored Traveling Tournament Problem* (mTTP) is an optimization problem that represents certain types of sports timetabling, where the objective is to minimize the total distance traveled by the teams. This work proposes the use of hybrid heuristic to solve the mTTP, using an evolutionary algorithm in association with the metaheuristic Simulated Annealing. It suggests the use of Genetic Algorithm with a compact genetic codification in conjunction with an algorithm to expand the code. The validation of the results will be done in benchmark problems available in literature and real benchmark problems, e.g. Brazilian Soccer Championship.

1 Introduction

Scheduling problems in sports has become an important class of optimization problems in recent years. The professional sport leagues represent a major economic activity around the world. For several sports, e.g. soccer, basketball, football, baseball, hockey, etc, where the teams plays a double round-robin tournament among themselves, where the games are played in different places during some time period, the automating of that schedulings are necessary and very important. Other facts fortify the application of optimization techniques. Teams and leagues do not want to waste their investments in players and structure in consequence of poor scheduling of games; sports leagues represent significant sources of revenue for radio and television world networks; the scheduling interfere directly in the performance of the teams; etc. On the other hand, sport leagues generate extremely challenging optimization problems, that attract attention of the Operational Research communities.

The scheduling problems in sports are known in the literature as Traveling Tournament Problem and it was proposed by Easton et al. [7]. The TTP abstract the salient features of Major League Baseball (MLB) in the United States and was established to stimulate research in sport scheduling. Since the challenge instances were proposed the TTP has raised significant interest. Several works in different contexts (see e.g. [2],[4],[9],[12],[13],[14],[18]) tackled the problem of tournament scheduling in different leagues and sports, which contains many interesting discussion on sport scheduling. Basically, the schedule of MLB is a

conflict between minimizing travel distances and feasibility constraints on the home/away patterns. A TTP solution is a double round-robin which satisfies sophisticated feasibility constraints (e.g. no more than three away games in a road trip) and minimizes the total travel distances of the teams.

Problems of that nature contain in general many conflicting restrictions to be satisfied and different objectives to accomplish, like minimize the total road trips of the teams during the tournament, one just game per day and per team, accomplishment of certain games in stadiums and in pre-established dates, number of consecutive games played in the team's city and out, etc. To generate good schedulings, satisfying all constraints, is a very hard task. The difficulty of solution of that problem is attributed to the great number of possibilities to be analyzed, e.g., for a competition with 20 teams there are $2,9062$ x 10^{130} possible combinations ([3]).

This work proposes the application of evolutionary techniques and local search for solving the mirrored version of the TTP, known as Mirrored Traveling Tournament Problem - mTTP ([13]). This paper is organized in five sections, being this the first. In the next section, the traveling tournament problem and your mirrored version are described. In section three, the methodology is detailed, with neighborhoods and the algorithm implemented. Section four presents the computational results and section five describe some conclusions of this paper.

2   Problem description

The *Traveling Tournament Problem* was first proposed by Easton et al. in [7]. A scheduling to a *double round-robin* (DRR) tournament, played by n teams, where n is a even number, consists in a schedule where each team plays with each other twice, one game in its home and other in your opponent's home. A game between teams T_i and T_j is represented by unordered pair (i, j). That schedule needs $2(n-1)$ rounds to represents all games of the tournament. The input data consists of the number of teams (n), a symmetric matrix D, n x n, where D_{ij} represents the distance between the home cities of the teams T_i and T_j.

The cost of a team is the total distance traveled starting from its home city and return there after the tournament ending. The cost of the solution is the sum of the cost of every team.

The objective is to find a schedule with minimum cost, satisfying the following constraints:

- No more than three consecutive home or away games for any team;
- A game of T_i at T_j's home cannot be followed by the game of T_j at T_i's home;

The *Mirrored Traveling Tournament Problem* (mTTP) proposed by Ribeiro and Urrutia in [13] is a generalization of TTP that represents the common structure in Latin-America tournaments (e.g. Brazilian Soccer Championship). The main difference is the concept of *mirrored double round-robin* (MDRR). A MDRR is a

tournament where each team plays every other once in the $n-1$ rounds, followed by the same games with reversed venues in the last $n-1$ rounds.

The objective is the same of TTP, find a schedule with minimum cost satisfying the same constraints plus an additional constraint: the games played in round R are the same played in round $R + (n - 1)$ for $R = 1, 2, ..., n - 1$, with reversed venues.

3 Methodology

The methodology used to solve the problem is based on the use of Genetic Algorithms in association with the metaheuristic Simulated Annealing. The idea is to use the Genetic Algorithm as construction phase, generating new solutions starting from the individuals' crossing and the Simulated Annealing to improve the local search in those new solutions.

The representation of a schedule is a table indicating the opponents of the teams, where each line corresponds to a team and each column corresponds to a round. The opponent's representation is given by the pair (i, j), where i represents the team T_i and j represents the round r_j (e.g., the opponent of the team T_1 in round r_2 is given by $(1, 2)$). If (i, j) is positive, the game takes place at T_i's home, otherwise at T_i's opponent home.

In this work only the $n - 1$ first rounds (*first half*) are represented, because the $n - 1$ last rounds (*second half*) are the mirror with reversed venues and all alteration in the first half affects the second half (see Figure 1).

T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 1. Representation of a Schedule

The rest of that section describes the neighborhood and the algorithms implemented in this work.

3.1 The Neighborhood

Three different moves have been defined to compose distinct kinds of neighborhood, named *Home-away swap*, *Team swap* and *Games swap*, from a schedule S. The neighborhood of a schedule S is the set of the schedules (feasibles and infeasibles) which can be obtained by applying one of three types of moves.

Home-away swap This move swaps the home/away roles of a game involving the teams T_i and T_j. The application of the move *Home-away swap* in a solution S obtain a solution S', with a single game swapped, by reversing the game's place. In other words, if team T_i plays at home with T_j (T_j plays away) in S, then T_j plays at home and T_i plays away in S'.

The figure 2 shows a schedule before and after the application of *Home-away swap* move. In this example, the move swap the home/away roles of the game involving the teams T_1 and T_4.

T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10
1	-6	-4	2	-5	-3	6	4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	1	6	2	-5	3	-1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 2. Schedule before (left) and after (right) the application of *Home-away swap*

Team swap This move swaps the schedule of two teams, T_i and T_j. Only the games where T_i and T_j play against each other are not swapped.

T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-3	-5	6	-4	-2	3	5
2	-3	5	-1	-4	-6	3	-5	1	4	6
3	2	-6	-5	1	4	-2	6	5	-1	-4
4	-5	-1	6	2	3	5	1	-6	-2	3
5	4	-2	3	6	1	-4	2	-3	-6	-1
6	1	3	-4	-5	2	-1	-3	4	5	-2

Fig. 3. Schedule before (left) and after (right) the application of *Team swap*

The figure 3 shows the application of *Team swap* move in teams T_3 and T_5.

Games swap This move consists in selecting an arbitrary game and enforcing it to be played in a round, followed by the necessary modifications to avoid teams playing more than one game in the same round. In consequence, more games would be swapped to maintain the feasible of the schedule. The modifications that have to be applied to the current schedule give rise to an ejection chain move.

Ejection chains are based on the notion of generating compound sequences of moves by linked steps in which changes in selected elements cause other elements to be ejected from their current state, position, or value assignment ([13]).

T_i/r_k	First half					Second half					T_i/r_k	First half					Second half				
	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3	1	-5	-3	2	4	-6	5	3	-2	-4	6
2	-5	3	-1	-4	-6	5	-3	1	4	6	2	-6	4	-1	-5	3	6	-4	1	5	-3
3	4	-2	-5	6	1	-4	2	5	-6	-1	3	-4	1	-5	6	-2	4	-1	5	-6	2
4	-3	-1	6	2	-5	3	1	-6	-2	5	4	3	-2	6	-1	5	-3	2	-6	1	-5
5	2	-6	3	1	4	-2	6	-3	-1	-4	5	1	-6	3	2	-4	-1	6	-3	-2	4
6	1	5	-4	-3	2	-1	-5	4	3	-2	6	2	5	-4	-3	1	-2	-5	4	3	-1

Fig. 4. Schedule before (left) and after (right) the application of *Games swap*

The figure 4 shows the schedule produced by the application of *Game swap* move. Note that several games are swapped, when the team T_3 was enforcing to play with team T_1 in round r_2.

3.2 The Algorithm

In this work was implemented a Genetic Algorithm ([10]) that uses the Simulated Annealing ([11]) metaheuristic to address new individuals to a local optimum. The application of local search in the individuals can be related with the combination of learning and evolution (Baldwin effect, [17]). In general, the learning is a search for the near viable solution and the modifications will be incorporate for the individual. The use of the SA metaheuristic leaves the stage of local search more aggressive, resulting in individuals more and more adapted inside of the population.

A compact representation of the chromosomes (individuals) was proposed for the application of the GA. The chromosomes are submitted to an algorithm of code expansion, which decodes them in scales of games.

In this representation, each gene of the chromosomes is associated to a team. The figure 5 presents an example of the compact representation used. Once the

$$3\ 2\ 5\ 1\ 6\ 4$$

Fig. 5. Example of a chromosome for a tournament with 6 teams

chromosome is defined, then it is submitted to an algorithm of code expansion, the well known Polygon Method ([5]), in association with heuristic to definition of the home/away roles. All solutions generated by this phase are feasible.

Consider a vector V of size n (n even) where each position i is associated to a team. The execution of the polygon method starts with the definition of the base team. This team is positioned in the first position of the vector V. The other teams are positioned in the rest of the positions, $i = 2, .., n$. In each round $r_i = 1, ..., n - 1$ the base team plays with the team of the position 2. The teams of positions $i = 3, ..., (n/2) + 1$ plays with the teams of positions $n - i + 3$ (n is the number of teams). Defined all games of a round r_i, the teams of position $i = 3, ..., n$ are moved to the position $i - 1$ and the team of position 2 are moved to position n. The base team are not moved. The figure 6 presents an example of execution.


Fig. 6. Polygon Method

The initial population is randomly defined and each individual is submitted to a *Randomized Non-Ascendent* method (RNA), with the *home-away* swap move only .

A reproduction mechanism, based on evolutionary processes, is applied on the population to explore the search space and to find better solutions for the problem. The operator of crossover implemented was based on the Block Order Crossover (BOX, [15]), illustrated in the Figure 7 . The "parents" are combined, through the random copy of both individuals blocks, what results in a new offspring, containing the two parents' genetic information.
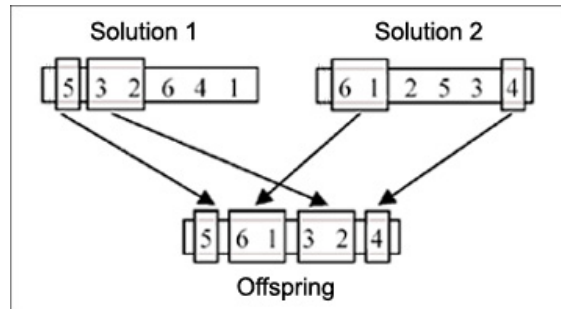
Fig. 7. BOX Crossover

In each generation a constant number of individuals are selected. Two individuals are randomly selected in the population for each crossover, which produces only one new individual. This offspring can, eventually, suffer mutation. The mutation used was the *Games Swap* move (see fig. 4).

Immediately after the execution of the genetic operators, the algorithm of code expansion is executed for generation the schedule.

The figure 8 presents the pseudo code of the algorithm implemented.

```
procedure GA
1  Initialize Population P;
2  while (g < nGenerations) do
3     while (i < nIndividuals) do
4        parents ← Select two individuals ∈ P;
5        offspring ← Crossover(parents);
6        if (MutationAcceptionCriterion) then
7           Mutation(offspring);
8        end-if
9        offspring' ← LocalSearch(offspring);
10       Evaluate(offspring');
11       Add offspring' in P;
12       i ← i + 1;
13    end-while
14    P ← Select(P);
15    g ← g + 1;
16 end-while
end-GA
```

Fig. 8. Pseudo Code of the implemented GA

The metaheuristic Simulated Annealing, with *Home-away swap* and *Team swap* moves, are applied in each offspring, to address it to a local optimum.

The idea of use the methods above mentioned in association is related to the fact that evolutionary algorithms find some difficulties to treat optimization problems with constraints. On the other hand, isolated application of optimization techniques based in constraints and local search can find more difficulties in virtue of low quality local optimum.

The application of evolutionary algorithms together with local search is justified for two main aspects:

– The evolutionary process will act just in the compact codification, always feasible. Therefore not reducing your effectiveness;
– The existence of a candidates population for the solution in evolution, together with the application of a local search procedure, increases the chances of obtaining local optimum of good quality.

4 Experiments and computational results

The algorithm was coded in C++ and was run on *Pentium IV* 3.0 *GHz clock with* 512 *Mbytes of RAM memory.*

The benchmark's instances, described in [7] and adapted to the mirrored form in [13] was used to validate the results. A real-life instance (br2003.24), where 24 teams playing in the main division of the 2003 edition of Brazilian Soccer Championship was also tested. These test problems are available to download in http://mat.gsia.cmu.edu/TOURN/. The parameters of the methods were empirically chosen, after several simulations. The population starts with 100 individuals and 50 offsprings are generated in each one of the 10 generations. The mutation probability and crossover probability are 30% and 100% respectively.

Table 1 shows the results for the considered instances. For each instance is reported the best known solution ([13]), the obtained by this approach, the relative gap in percent between the best and obtained solutions, and the total computation times in seconds.

Table 1. Computational results

Instances	Best known	Obtained	$gap(\%)$	Time(*secs.*)
circ4	20	20	0%	2
circ6	72	72	0%	4
circ8	140	142	1,41%	48
circ10	240	282	14,89%	365
circ12	456	458	0,44%	51
circ14	714	714	0%	26
circ16	980	1014	3,35%	264
circ18	1306	1370	4,67%	604
circ20	1882	1890	0,42%	28
nl4	8276	8276	0%	2
nl6	26588	26588	0%	3
nl8	41928	43112	2,75%	55
nl10	58190	66264	12,18%	130
nl12	120655	120981	0,27%	317
nl14	208086	208086	0%	140
nl16	279618	290188	3,64%	142
br2003.24	503158	511256	1,58%	938

The results presented demonstrate that the proposed methodology can be competitive, because using only three movements was possible to obtain values near of the best results known in the literature, getting to reduce to zero the gap in six of the seventeen instances and being near of reducing to zero in nine others. On the other hand, the results of circ10 and nl10 wasn't good, demonstrating the requirement of improving the algorithm, mainly the neighborhoods. This fact is

easily explained. The schedule generated by this implementation is fixed in a pattern, due the polygon method characteristics. The ejection chains moves are very important, because they are able to find solutions that are not reachable through other neighborhoods. Admissible moves in other neighborhoods may appear after an ejection chain move is performed, in situations where none existed before. If they are not used, algorithms based on local improvement strategies may easily stop at local optimum of low quality.

The result of the real-life instance was very interesting. First, because it is the larger instance in the literature, with 24 teams. Second, because was observed a reduction of 51.2% in the total distance traveled, where in the official schedule the teams traveled 1.048.134 km and in the schedule found by the work they traveled 511.256 km only.

5   Conclusions

In this work was investigate the Mirrored Traveling Tournament Problem, first published in [13], with a implementation of a Genetic Algorithm in association with the metaheuristic Simulated Annealing.

A compact representation for the chromosomes (population individuals) was proposed. These chromosomes are submitted to a code expansion algorithm, used to decodes them in scales of games. Initially, the home-away roles are aleatory defined. After this phase a *Randomized Non-Ascendent* method (RNA) is applied in the schedule to address it to a local optimum. This method showed effectiveness, because all solutions found are feasible, increasing the chance of obtained local optimum with good quality.

Three different neighborhood structures for local search was investigated: two simple neighborhood (*Home-away swap* and *Team swap*) and a more complicated based on ejection chain, whose the importance was described in section three. The results show the need of explores other neighborhoods to escape from the pattern imposed to the solutions generated by the Polygon Method.

The approach become very promising, when the reported results are analyzed. Seventeen benchmark instances was tested and in six the algorithm got to reduce to zero the gap and almost reduced to zero in nine others. One real-life instance, the 2003 edition of Brazilian Soccer Championship, was also tested, with reduction of 51.2% of the total distance traveled by the official schedule.

Finally, this work explores the mirrored instances of TTP, because its represents common structure in Latin-America tournaments (e.g. Brazilian Soccer Championship). There are a variety of open issues that need to be addressed, e.g., to consider a large neighborhood to obtain high-quality solutions. In the same way the championships has many other restrictions that should be analyzed: treatment of the classic games, allocation of stadiums, do not consider metric distance, but the airfares, and many others real constraints.

References

1. Anagnostopoulos, A., Michel, L., Van Hentenryck, P., Vergados, Y.: A Simulated Annealing Approach to the Traveling Tournament Problem. Proceedings of Cpaior'03 (2003)
2. Biajoli, F. L., Chaves, A. A., Mine, O. M., Souza, M. J. F., Pontes, R. C., Lucena, A., Cabral, L. F.: Scheduling the Brazilian Soccer Championship: A Simulated Annealing Approach. Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat2004, Pittsburgh, USA, (2004) 433-437
3. Concílio, R., Zuben, F. J.: Uma Abordagem Evolutiva para Geração Automática de Turnos Completos em Torneios. Revista Controle e Automação. Vol. 13, n.2, (2002) 105-122
4. Costa, D.: An Evolutionary Tabu Search Algorithm and the NHL Scheduling Problem. Infor. Vol. 33, n. 3, (1995) 161-178
5. Dinitz, J., Lamken, E., Wallis, W. D.: Scheduling a Tournament. In C.J. Colbourn and J. Dinitz, editors, Handbook of Combinatorial Designs, CRC Press, (1995) 578-584
6. Dowsland, K. A.: Simulated Annealing. C.R. Reeves, Editor, Modern Heuristic Techniques for Combinatorial Problems, Advanced Topics in Computer Science Series, Chapter 2. Blackwell Scientific Publications, London, (1993) 20-69
7. Easton , K., Nemhauser, G., Trick, M.: The Traveling Tournament Problem Description and Benchmarks. Seventh International Conference on the Principles and Practice of Constraint Programming, CP'99, (2001) 580-589
8. Goldberg, D. E.: Genetic Algorithms in Search. Optimization and Machine Learning. Addison-Wesley: Berkeley, 223 pag., (1989)
9. Hamiez, J. P., Hao, J. K.: Solving The Sports League Scheduling Problem with Tabu Search. Lecture Notes In Artificial Intelligence 2148, Springer (2001), 24-36
10. Holland, J. H.: Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 211 pag., (1975)
11. Kirkpatrick, S., Gellat, D. C., Vecchi, M. P.: Optimization by Simulated Annealing. Science, Vol. 220, (1983) 671-680
12. Nemhauser, G., Trick, M.: Scheduling a Major College Basketball Conference. Operations Research, Vol. 46(1), (1998) 1-8
13. Ribeiro, C.C., Urrutia S.: Heuristics for the Mirrored Traveling Tournament Problem. Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat2004, Pittsburgh, USA, (2004) 323-342
14. Schönberger, J., Mattfeld, D. C., Kopfer, H.:Memetic Algorithm Timetabling for Non-Commercial Sport Leagues. European Journal of Operational Research, Vol. 153(1), (1989) 102-116
15. Syswerda, G.: Uniform Crossover in Genetic Algorithms. International Conference on Genetic Algorithms (ICGA), Vol. 3, Virginia, USA,(1989) 2-9
16. Trick, M. A.: A Schedule-Then-Break Approach to Sports Timetabling. Lecture Notes In Computer Science, Vol. 2079, (2000) 242-253
17. Whitley, D., Gordon, V. S., Mathiask K.: Lamarckian Evolution, the Baldwin Effect and Function Optimization. In Davidor, Y., Schwefel, H.-P. and Männer, R. (eds.) Proceedings of the Third Conference on Parallel Problem Solving from Nature, BerlSpringer, October, (1994) 6-15
18. Zhang, H.: Generating College Conference Basketball Schedules by a Sat Solver. Proceedings Of The Fifth International Symposium on the Theory and Applications of Satisfiability Testing, Cincinnati, (2002) 281-291