

A Sport Tournament Scheduling by Genetic Algorithm with Swapping Method

Tinnaluk Rutjanisarakul and Thiradet Jiarasuksakun
Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT),
10140 Bangkok, Thailand

Abstract: A sport tournament problem is considered the Traveling Tournament Problem (TTP). One interesting type is the mirrored Traveling Tournament Problem (mTTP). The objective of the problem is to minimize either the total number of traveling or the total distances of traveling or both. This research aims to find an optimized solution of the mirrored traveling tournament problem with minimum total number of traveling. The solutions consisting of traveling and scheduling tables are solved by using Genetic Algorithm (GA) with swapping method. The numbers of traveling of all teams from obtained solutions are close to the lower bound theory of number of traveling. Moreover, this algorithm generates better solutions than known results for most cases.

Key words: Sport tournament, traveling tournament problem, mirrored traveling tournament problem, minimum total number of traveling, objective, interesting

INTRODUCTION

There are many types of graph tournaments such as round robin tournament, double round robin tournament, single-elimination tournament, double-elimination tournament, etc. The differences among these tournaments are scheduling, number of matches and format of the competitions. For example, the FIFA world cup in the final tournament has two stages: group stage and knockout stage. In the group stage, there are eight groups and each group contains four teams. Each group plays a round robin tournament in which each team plays one game against every other team within the group. Then only top two teams in each group would go to the knockout stage. The knockout stage is a single-elimination tournament in which teams play against each other in one-off matches. Another example is the tennis competition consisting of four most important annual grand slam championships. The differences among these championships are the weathers and court types but all championships are single-elimination tournament in which the winner must win seven games consecutively. However, almost all other sport leagues, e.g., soccer, basketball, football, baseball, etc. are double round robin tournament in which each team plays against each other team in two games, one as home game and another one as away game.

The scheduling problem of soccer league is more popular than other sports scheduling problem because most countries have formed their own soccer leagues. Each soccer league has usually organized a double round

robin tournament. Each playing team has to find sponsors to support their costs for a tournament such as advertisement, traveling and accommodation. Therefore, it would be beneficial to all if the tournament organizer could help to reduce some costs. All teams do not want to waste their time and money in transportation as a consequence of poor scheduling of the games. In a sport league, there are many variables affecting the costs of each team such as means of traveling, distances of traveling, number of traveling, sequences of traveling, etc. Thus, the tournament scheduling problem affecting the traveling cost has become an important class of optimization problems for many years.

The tournament in graph theory is an orientation of a complete graph used to describe the result of tournament in finding the winner of the tournament. An arc in the directed graph is used to show the result of each pair of teams in the tournament (Chartrand and Zhang, 2005). The scheduling problems in sports are known in the literatures as traveling tournament problem proposed by Easton *et al.* (2001). Many educators were interested in scheduling problem and aimed to study the Traveling Tournament Problem differently (Ahmed, 2004; Bijaoli and Lorena, 2006; Ribeiro and Urrutia, 2007). They focused on improving method for solving Traveling Tournament Problem such that their results in scheduling were better approximation than preceding methods such as integer programming, canonical schedules found by Hamiltonian cycles, etc. Mostly, their objective is to minimize the total distance traveled by all teams. They generated the schedule of tournament firstly and then

they applied their programs to find the minimum total distance of all teams. The mirrored traveling tournament problem with new objective which is to minimize the longest distance traveled is proposed by De Carvalho and Lorena (2012). Urrutia and Ribeiro (2006) studied the mirrored traveling tournament problem and found the maximizing breaks and bounding number of traveling of solutions for each number of teams. In this research, the objective is to minimize the total number of traveling of all teams by using Genetic algorithm with swapping process. The optimized traveling tournaments of all teams are formed firstly. Next the scheduling of tournaments of all teams are generated which suitable to the optimized traveling tournament. The swapping process was used to apply the crossover process in Genetic algorithm. Moreover, the obtained solutions were compared with the known results and the bounding number of traveling (Urrutia and Ribeiro, 2006).

MATERIALS AND METHODS

Problem description: The number of traveling of each team is the total number of traveling starting from its home city and return there after the end of tournament. A game at its home city is called home game and a game at the opponent’s city is called away game. Each team has at most one game in a week and each week must have equal number of home games and away games.

The objective is to find a tournament with minimum total number of traveling of all teams, satisfying the following constraints:

- The tournament scheduling here is the mirrored Traveling Tournament Problem (mTTP)
- The tournament problem here consists of 2 parts: team traveling and scheduling. The traveling of tournament refers to traveling of all teams. The scheduling of tournament refers to scheduling of all games of every team
- Each team does not go back to its home city after away game if it does not have a home game in a week after
- TTP’s condition: The traveling sequence of each team is allowed to have at most three consecutive home games or three consecutive away games

Notations: In this research, the tournament problem is separated into two parts: team traveling and team scheduling. The team traveling and scheduling of n team tournament are represented by tables A and B, respectively where n is even. The team traveling table A consists of n traveling sequences of each n teams.

The traveling sequence of team i is a sequence of 0’s and 1’s where 0 stands for home game and 1 stands for

Table 1: One possible traveling sequence of team 1

Team	$w_1, w_2, \dots, w_{n/2}$	$w_{n/2+1}, \dots, w_n$
1	0 1, ..., 0	1 0, ..., 1

Table 2: An example of team traveling for four team tournament

Team (A ⇒)	w_1	w_2	w_3	w_4	w_5	w_6
1	0	0	0	1	1	1
2	1	0	0	0	1	1
3	0	1	1	1	0	0
4	1	1	1	0	0	0

Table 3: One possible scheduling of team 1

Team	w_1	w_2	w_3	w_4	w_5	w_6
1	2	3	4	2	3	4

Table 4: An example of team scheduling for four team tournament

Team (B ⇒)	w_1	w_2	w_3	w_4	w_5	w_6
1	2	3	4	2	3	4
2	1	4	3	1	4	3
3	4	1	2	4	1	2
4	3	2	1	3	2	1

away game. The element in the table A(i, j) represents a type of game of team j on week w_j . For example, one possible traveling sequence is shown in Table 1.

From the problem description, each traveling sequence may have at most three consecutive 0’s or three consecutive 1’s. After all traveling sequences are obtained, the team traveling can be formed. An example of four team tournament is shown in Table 2.

It is clear to see that any pair of teams has different traveling sequences. Next, the team scheduling also consists of n scheduling sequences. The scheduling sequence in table B of team i is a sequence of number from 1 to n except number i. The element B(i, j) represents team number playing against team i on week w_j . For example, one possible scheduling sequence for four team tournament as shown in Table 3.

From Table 3, it is a scheduling sequence of team 1 and there is no number 1 in team 1’s scheduling sequence. In general, there is no number 1 in team i’s scheduling sequence. After all scheduling sequences are obtained, the team scheduling can be formed. An example of team scheduling for four team tournament is shown in Table 4. From Table 4, element in each B(i, j) depends on the team traveling in Table 2. For example, B(1, 3) can be equal to 4 because A(1, 3) is not equal to A(4, 3). In general, B(i, j) can be equal to k if and only if A(i, j) is not equal to A(k, j), since, for each game, one team plays a home game and another team plays an away game.

Moreover, the sport tournament problem is solved by using the Genetic algorithm together with a swapping method. A swapping method is used to generate a new traveling sequence. This method swaps the home/away roles of all games in a traveling sequence. Figure 1 shows the traveling sequences before and after one swapping.

Before:	1	0	0	0	1	1
After:	0	1	1	1	0	0

Fig. 1: Traveling sequences before and after one swapping

Table 5: Generating the first three traveling sequences for six team tournament

Team	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
1	0	0	1	1	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0	1	0
3	1	0	1	1	0	0	1	0	0	1

Table 6: A complete team traveling for six team tournament

Team	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
1	0	0	1	1	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0	1	0
3	1	0	1	1	0	0	1	0	0	1
4	1	1	0	0	1	0	0	1	1	0
5	0	1	0	1	0	1	0	1	0	1
6	0	1	0	0	1	1	0	1	1	0

In addition, an individual in GA is found by generating n/2 traveling sequences first. Then another n/2 traveling sequences are generated by using a swapping method. An example of generating an individual in case six teams is made by the following step. Firstly, the first three traveling sequences are generated by randomly choose with the TTP's condition. An example of this step is shown in Table 5.

After that the remaining traveling sequences of team 4-6 are generated by using a swapping method with the traveling sequences of team 1-3, respectively. An example is shown in Table 6.

Algorithm: The algorithm here bases on the Genetic algorithm that has mutation and crossover processes. This algorithm is used to find the optimization solution of the team traveling but the team scheduling is found by another algorithm called "Schedule." By deleting row and column heads of table A, we obtain the n×2 (n-1) matrix before applying the following algorithm.

In the first algorithm, the mutation is performed before crossover by randomly choosing a week from 1 to n-1 of a random team to change from either 1-0 or 0-1. After that a mirrored week is changed by the same condition due to mirrored tournament property. Moreover, two weeks of swapping teams are also changed by swapping property. An example of mutation process is shown in Fig. 2.

Next, the crossover process is used to generate a new individual (team traveling) from two source individuals by the following steps:

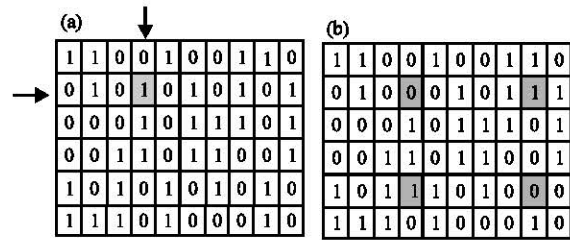


Fig. 2: The mutation process: a) Before mutation process and b) After mutation process

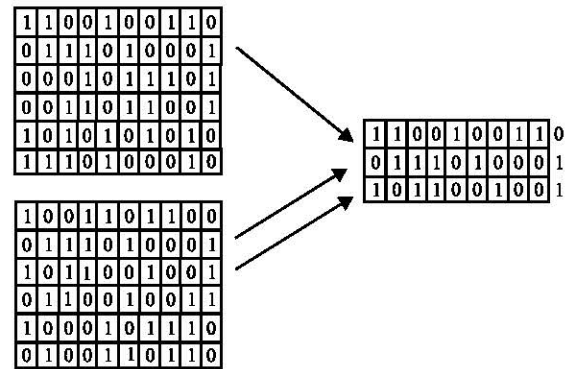


Fig. 3: A crossover process: a) Solution A; b) Solution B and c) Solution C

1	1	0	0	1	0	0	1	1	0
0	1	1	1	0	1	0	0	0	1
1	0	1	1	0	0	1	0	0	1
0	0	1	1	0	1	1	0	0	1
1	0	0	0	1	0	1	1	1	0
0	1	0	0	1	1	0	1	1	0

Fig. 4: A new solution

- Find the ratio of each source individual to define a number of randomly chosen traveling sequences
- Randomly choose the first n/2 traveling sequences of a new individual depending on previous step
- After the first n/2 traveling sequences are obtained, the n/2 remaining traveling sequences are generated by using a swapping method

Figure 3 shows the crossover process of two individuals in case of six team tournament. After the crossover process, the swapping process is used to generate remaining teams of a new solution as shown in Fig. 4.

After the optimized solution is obtained, "Schedule" algorithm is used to find the schedule of all teams. Alghoriyhm 1 presents the pseudo code of "Schedule" algorithm.

Table 7: First half of traveling sequences of team 1 and team 2 for six team tournament

Team	w ₁	w ₂	w ₃	w ₄	w ₅
1	1	0	1	1	0
2	1	0	0	1	1

Table 8: First step in making a scheduling sequence fo team 1 for six team tournament

Team	w ₁	w ₂	w ₃	w ₄	w ₅
1			2		

Table 9: First half of a scheduling sequence of team 3 for six team tournament

Team	w ₁	w ₂	w ₃	w ₄	w ₅
3	4	1	0	2	6

Algorithm 1: Pseudo code of “Schedule” algorithm

```

Produce schedule (B [1:n, 1: 2 (n-1)])
1 Receive the optimized travelling of all teams (A [1: n, 1: n-1])
2 for (i = 1: i = n)
3   if (i>1) then
4     add schedule with team before team i
5   end = if
6   for (m = 1: m = n-1)
7     if (B [i, m] satisfy conditions) then
8       add schedule B [i, m]
9     end-if
10    while min (B [i, :] = 0)
11      nteam-missingteam (B [i,m])
12    if (nteam == i+1) then
13      edit schedule team i-1
14    else
15      edit schedule team I
16    end-if
17  end-while
18 end-for
19 end-for
20 B [1:n, n:2n-2]-B [1:n, 1:n-1]
end-produce
    
```

The schedule algorithm has two mainly subroutines: “Add schedule” and “Edit schedule”. This algorithm considers only the first half of traveling sequences of all teams. First, the algorithm adds schedule from team 1 to team n by comparing traveling of 2 teams. For example, if number 2 is added in scheduling of team 1 for six team tournament then the algorithm compares traveling of team 1 and team 2 as shown in Table 7.

There are two weeks (2 columns) with different numbers. It means that number 2 can be assigned in week 3 or 5 in scheduling of team 1. However, the algorithm is designed to choose the smaller week number. So, the number 2 is added in week 3 as shown in Table 8.

After the number 2 is added in scheduling of team 1, the algorithm repeats this process with team 3, 4 and n, respectively. However, the algorithm sometimes cannot add some team numbers in current scheduling. The number 0 is then added instead in remaining weeks. For example, scheduling of team 3 in six team tournament does not have team number 5 as shown in Table 9. The number 0 then appears instead of number 5 in the scheduling sequence. So, the algorithm uses another subroutine called “Edit schedule” to edit this schedule until it can add all numbers satisfying all tournament’s conditions.

Table 10: The computational results

n	OR	LB	KR	Gap#1 (OR-LB)	Gap#2 (OR-KR)
4	17	17	17	0	0
6	48	48	48	0	0
8	80	80	80	0	0
10	130	130	130	0	0
12	192	192	192	0	0
14	253	252	256	1	-3
16	348	342	342	6	6
18	432	432	434	0	-2
20	521	520	526	1	-5

RESULTS AND DISCUSSION

The algorithm in this resaerch starts with four randomly populations. In each iteration, the two best solutions are kept and another two solutions are found by the mutation process is used with only 80% probability. However, the crossover process is always used.

Table 10 shows the results of total numbers of traveling for some even numbers of team n. The Obtained Results (OR) are compared with the Known Results (KR) and the Lower Bounds (LB) of the mirrored traveling tournament problem (Urrutia and Ribeiro, 2006). In addition, there are two relative gaps shown in Table 10. The first relative gap (gap#1) is the difference between the obtained results and the lower bounds. The second gap (gap#2) is the difference between the obtained results and the known results.

Thus, Table 10 indicates that most obtained results approach to the lower bounds in the literature and they are better than the known results.

CONCLUSION

In this resaerch, we develop the Genetic algorithm model by adding the swapping process in the crossover process to generate another traveling sequence that has the same or close to the number of traveling of source. The result also shows that the difference of number of traveling between any two teams is at most two.

In addition, the swapping process can reduce the running times because the algorithm randomly chooses n/2 traveling sequences from the two individuals. Then the remaining traveling sequences are generated by using the swapping process. The obtained solution also satisfies the TTP’s conditions. However, the algorithm without the swapping process must randomly choose n traveling sequences from the two individuals and the obtained solution sometimes does not satisfy the TTP’s conditions. It means that the algorithm must repeat the crossover process until the solution satisfies the TTP’s conditions.

Moreover, a fair tournament defined by each team has the same number of traveling could be one of many constraints. This algorithm could make a fair tournament

in case of some number of team n . Nevertheless, a case of minimum number of traveling may not be the fair tournament.

ACKNOWLEDGEMENT

The researcher would like to thank Human Resource Development in Science Project (Science Achievement Scholarship of Thailand, SAST) for financial support.

REFERENCES

- Ahmed, Z.K., 2004. Genetic algorithm for the Traveling salesman problem using sequential constructive crossover operator. *Int. J. Biomet. Bioinform.*, 3: 96-96.
- Biajoli, F. and L. Lorena, 2006. Mirrored Traveling Tournament Problem: An Evolutionary Approach. In: *Advances in Artificial Intelligence -IBERAMIA-SBIA 2006*, Sichman, J.S., H. Coelho and S.O. Rezende (Eds.). Springer, Berlin, Germany, ISBN:978-3-540-45462-5, pp: 208-217.
- Chartrand, G. and P. Zhang, 2005. *Introduction to Graph Theory*. McGraw-Hill, New York, USA., ISBN:9780072948622, Pages: 449.
- De Carvalho, M.A.M. and L.A.N. Lorena, 2012. New models for the mirrored traveling tournament problem. *Comput. Ind. Eng.*, 63: 1089-1095.
- Easton, K., G. Nemhauser and M. Trick, 2001. The Traveling Tournament Problem Description and Benchmarks. In: *Principles and Practice of Constraint Programming*, Walsh, T. (Ed.). Springer, Berlin, Germany, ISBN: 978-3-540-42863-3, pp: 580-584.
- Ribeiro, C.C. and S. Urrutia, 2007. Heuristics for the mirrored traveling tournament problem. *Eur. J. Oper. Res.*, 179: 775-787.
- Urrutia, S. and C.C. Ribeiro, 2006. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Appl. Math.*, 154: 1932-1938.