ISSN 2816-8089

RESEARCH ARTICLE

A Comparative Analysis of Application of Genetic Algorithm and Particle Swarm Optimization in Solving Traveling Tournament Problem (TTP)

Avijit Haldar[1], Shyama Mondal[1], Alok Mukherjee[1*], Kingshuk Chatterjee[1]

[1]*Government College of Engineering and Ceramic Technology, Kolkata 700 010, India.*

Abstract

Traveling Tournament Problem (TTP) has been a major area of research due to its huge application in developing smooth and healthy match schedules in a tournament. The primary objective of a similar problem is to minimize the travel distance for the participating teams. This would incur better quality of the tournament as the players would experience least travel; hence restore better energy level. Besides, there would be a great benefit to the tournament organizers from the economic point of view as well. A well-constructed schedule, comprising of diverse combinations of the home and away matches in a round robin tournament would keep the fans more attracted, resulting in turnouts in a large number in the stadiums and a considerable amount of revenue generated from the match tickets. Hence, an optimal solution to the problem is necessary from all respects; although it becomes progressively harder to identify the optimal solution with increasing number of teams. In this work, we have described how to solve the problem using Genetic algorithm (GA) and particle swarm optimization (PSO).

Key Words: *Traveling Tournament Problem (TTP); Genetic algorithm (GA); Particle Swarm Optimization (PSO); Unconstrained TTP*

****Corresponding Author***: *Alok Mukherjee, Government College of Engineering and Ceramic Technology, Kolkata 700010, India; E-mail: alokmukherjee.ju@gmail.com*

Received Date: *August 17, 2022,* ***Accepted Date***: *August 26, 2022,* ***Published Date***: *August 29, 2022*

Citation: *Avijit Haldar, Shyama Mondal, Alok Mukherjee and Kingshuk Chatterjee. A Comparative Analysis of Application of Genetic Algorithm and Particle Swarm Optimization in Solving Traveling Tournament Problem (TTP). Int J Bioinfor Intell Comput. 2022;1(2):142-151.*

1. Introduction

The Traveling Tournament Problem (TTP) is a timetable scheduling problem related to sports tournaments. Professional sports leagues are a part of almost every country. Apart from the entertainment it provides to the people of the country, as well as of the whole world by the means of television and the internet, these tournaments often possess a vital share of a country's economic scenario. These leagues are able to gather very large revenues by selling the tickets, settling contracts for live streaming or broadcasting of the matches. Hence, scheduling the matches, i.e., preparing the tournament timetable, including the order of the games for each of the participating teams, as well as, scheduling the venues becomes vital for conducting a successful tournament [1]. This growing importance of TTP has made it an integral part of research in theoretical computer science, especially due to its hardness in reaching an optimal solution. TTP is intended to reduce the distance travelled in a double round-robin tournament (DRRT) where n teams are playing as a part of it. TTP is a kind of combinatorial optimization problem that combines features of two major problems related to theoretical computer science: the traveling salesman problem and the tournament scheduling problem or the vehicle routing problem; although, TTP is many folds harder than the traveling salesman problem. Even a tournament schedule containing a very small number of participating teams becomes extremely difficult to solve.

TTP is extremely important in the field of the tournament schedule. TTP is primarily intended to design a tournament schedule to minimize the total traveling distance. This is more important to reduce the distance travelled by the players, as well as for reducing the associated time of travel and minimize additional fatigue among the players. Hence, TTP holds the position of the most fundamental issues involved in developing a tournament schedule in a sports league, especially where the travel duration is an issue. The design of an optimal TTP is more important especially in such a host country where the land topology contains major diversities including hilly regions, large water bodies, deserts, or other diverse geographic terrains since the traveling to different points becomes extremely tiresome for the players, as well as costly for the organizing bodies. Apart from this issue of preventing too much travel of the participating teams, another major issue holds a major hindrance to the design of the problem. This is interpreted as the feasibility issue considering the home and away pattern. Lengthy sequence of either home or away matches is not desired; hence, care is required to be taken to avoid such scheduling and a fair combination of home and away matches has to be prepared. Constraint programming is instrumental mainly in solving complex sets of patterns related to home and away from constraints; whereas, integer programming is used to solve primarily the problems related to large traveling salesman problems, as well as vehicle routing problems to minimize travel distance [2]. Thus, TTP has emerged as a major problem involving research with a combined approach. Hence, an optimal solution with minimum travel duration is extremely important to maintain a better fitness level among the players, which in turn, helps in retaining the quality of the tournament.

The concept of TTP was brought to light by Easton, Nemhauser and Trick [3], where the authors presented a scheduling scheme of double round-robin tournament to minimize travel distance. The NP-completeness proof for a TTP variant has been proposed by Bhattacharyya [4]. Several other works have investigated other methods of analysis such as tabu search, integer programming etc. and other artificial intelligence-based techniques to obtain optimized solutions for TTP [5]. Constraint programming-based works have also been investigated by researchers [6,7], where the authors of [7] have studied different advances of constraint programming regarding scheduling problems. Round robin scheduling problems regarding TTP has been investigated by [8,9]. Some other researchers have paid attention towards repetition of home and away games [10]. Mirrored Traveling Tournament Problem (mTTP) is another challenging problem in the field of combinatorial optimization, which has been investigated by several researchers [5,11]. The authors of [12] have also investigated the mTTP and proposed a two-stage method regarding the same. A hybrid heuristic regarding mTTP has been proposed by the authors of [13] using clustering Search approach. The authors of [14] have studied the effect of breaks and distances of the TTP, where the authors have led to the analysis of distance minimization and breaks maximization. A different approach of Iteration Limited Threshold Accepting (ILTA) has been proposed by the authors of [15], where a new hyper-heuristic has been proposed by the authors. Another iteration-based model, such as Iterated Local Search (ILS) heuristic has been discussed in [16]. An effective encoding scheme has been proposed by the authors of [17], where, they have investigated several instances of TTP for double round robin TTP. Some of the other works have incorporated hybrid approach of Lagrangian relaxation and constraint programming [18]; whereas the authors of [19] showed improvement in the constraint of independent lower bound. Other works have also introduced factor approximation for reducing the search space [20].

In this work, we have proposed a method of solution to the Travelling Tournament Problem using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). After solving the problem with PSO and GA we can conclude that PSO is giving better results than GA and for more teams it is more significant.

2. Different Types of Problem

Let us consider n teams participating in a tournament, where n even. First, we shall consider a round-robin tournament, which is a class of tournament where each of the participating teams play each other in some order. Hence, such a round-robin tournament has $(n-1)$ slots, where the number of games played $n/2$. For each of the games, the team which plays at a venue closer to its hometown is denoted the home team. Thus, the opponent team is termed as the away team, as it plays the game away from its hometown. This arrangement differs from other situations where all teams travel to a single venue. A double round-robin tournament has $2\ (n-1)$ numbers of slots, and every pair of teams faces each other twice: once at the home venue and once at the away venue. This continues for each of the participating team. The distances of venues between the participating teams are given by an *n-by-n* matrix. This is called as distance matrix and is denoted by $[D]_{n \times n}$. When a team plays an away game, it is assumed to travel from its home site to the away venue. When a team plays consecutive away games, teams travel from one away venue to the next venue directly. Each team might begin playing the tournament playing a game at its hometown, i.e., as a home match. The same team must again return to home when it completes playing all the matches of the tournament. Consecutive away games demand for a large volume of road trips; whereas, back-to-back home games are kind of home stand. The length of a road trip or home stand is equal to the number of opponents played, although it is not the travel distance. Thus, TTP may be described as follows:

Input: Input will be the number of teams participating in the tournament, which is n; the distance matrix, as described earlier by $[D]_{n \times n}$, each element of which is an integer; L, U are integer parameters.

Output: Output will be the schedule of a double round robin tournament of the *n* number of participating teams. The length of every road trip and the home stand should lie between L and U inclusive. The objective of the schedule should be to minimize the total travel distance by the teams.

The above are the basic constraints of developing the schedule. Other additional requirements for the TTP solution may include the following:

Mirrored: In a double round robin tournament, there must be provision for a round robin tournament in the first *(n-1)* slots. This should follow another similar tournament with venues reversed in the second *(n-1)* slots.

No Repeaters: There are no teams *i*, *j* such that *i* plays at *j* and then *j* plays at *i* in the next slot. We will refer to these two variants as the Traveling Tournament Problem/Mirrored (TTP/Mirror) and the Traveling Tournament Problem/No Repeaters (TTP/No Repeat) respectively. The parameters L and U define the tradeoff between distance travelled and the length of the home stands and road trips.

3. Methodology

In this work, as mentioned before, two major bio and nature inspired, artificial intelligence-based optimization techniques, such as, genetic algorithm (GA) and particle swarm optimization (PSO) has been adopted to solve the designed TTP. The methodologies are described as below.

3.1. Genetic algorithm

A genetic algorithm (GA) is a major bio inspired optimization technique which has a large application in the field of research. This optimization technique is a search heuristic, which is developed from the inspiration of the theory of natural evolution, as proposed by Charles Darwin. This algorithm primarily deals with the process of natural selection, in which it is proposed that the fittest individuals will survive and will be selected from a large population for reproduction which will carry to the next generation. The offspring so produced, will inherit the characteristics of the parents. They will further carry and add these characteristics to the next generation. If the concerned parents have better fitness, the offspring so produced will show characteristics better than parents and naturally their chance of survival will be better.

This will be an iterative process and gradually as the evolution continues, generations with fitter individuals will be found; and finally, a generation with fittest individuals will evolve. This concept of developing the fittest individuals is applied to a search problem which is termed as Genetic Algorithm. Here, we consider a set of solutions for a problem and select the set of best ones. The five parameters which are involved in this process are as follows:

a) **Initial population:** Initially, a set of individuals is considered. This is termed as Population. Each individual may be a possible solution. Each such individual is characterized by a set of parameters. These parameters are variable in nature and are called Genes. Genes are gradually joined into a string. This string of genes is called

Chromosome, which is the solution to the problem. Usually, binary values, such as, strings of 1s and 0s, are used as genes. These genes are encoded in chromosomes. The orientation of Genes, Chromosomes and Population are explained graphically in Figure 1.

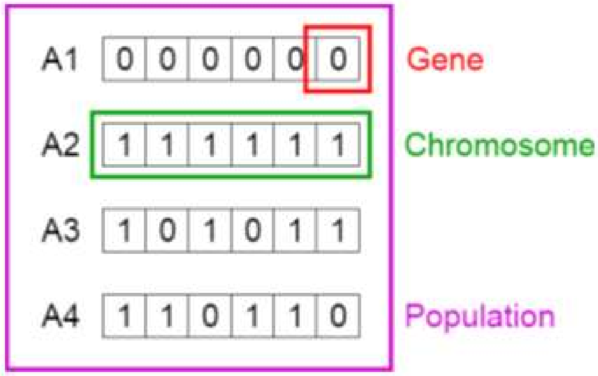

Figure 1: *Orientation of genes, chromosomes and population in genetic algorithm (GA).*

b) **Fitness Function:** The fitness function determines the ability of fitness of an individual. In other words, fitness function denotes the ability of an individual to compete with other individuals. It produces a fitness score for each individual. In GA, the probability of selection of an individual for reproduction is based on this fitness score so generated for that particular individual.

c) **Selection:** The idea of the selection phase is to identify the most suited individuals, i.e., the fittest individuals. This ensures that these fittest individuals pass their genes to the next generation. This allows selecting two pairs of individuals. This selection is again based on the fitness scores. These individuals are denoted as parents. Therefore, in a word, individuals who have the highest fitness scores are more acceptable for selection for reproduction.

d) **Crossover:** Crossover is the most significant phase among all the other steps of genetic algorithm. For each pair of parents to be met, a crossover point is chosen from within the genes of the two individuals at random. This is illustrated in Figure 2 where we have shown a typical example of crossover between genes of parents. This entire process of crossover and development of new offspring is illustrated in this figure, where the crossover point is considered to be the third element. Exchange of the genes of the parents continues among the genes themselves until the crossover point is reached. Thus, new offspring are created.

e) **Mutation:** Some of the genes of the new offsprings so formed might be mutate among themselves; although with a low random probability. In other words, it means that some of the bits in the bit string might be flipped. This is shown in Figure 3. Mutation is very much essential in order to maintain diversity within a certain set of population. This further prevents premature convergence.

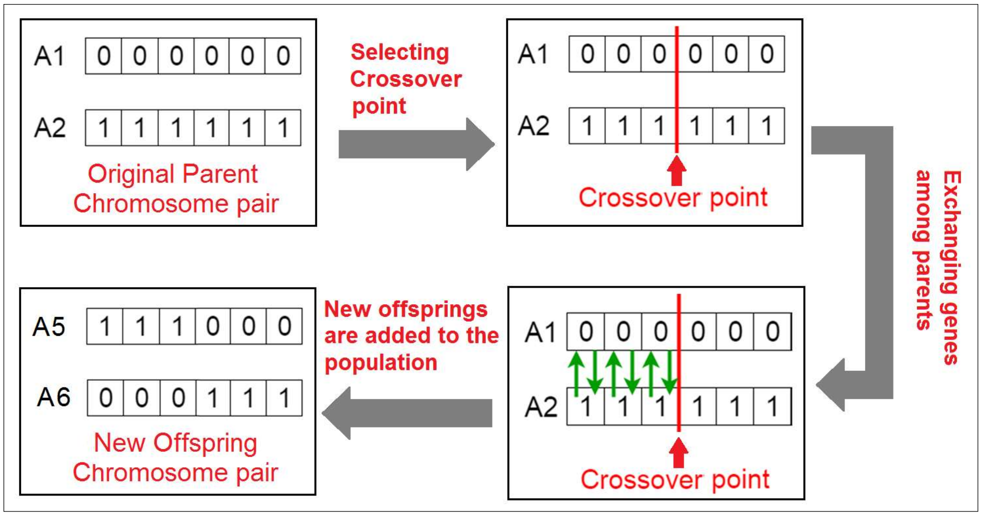

Figure 2: *Process of crossover and development of new offspring.*

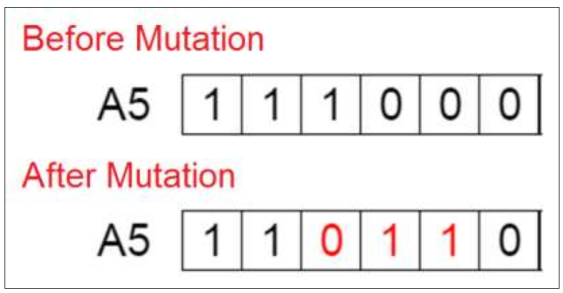

Figure 3: *Effect of mutation.*

3.1.1. Steps of genetic algorithm-based study

The steps associated with designing the proposed genetic algorithm-based analysis is described here in sequence:

1. Randomly initialize population p. This is described with an example here.

   Let BACDBDCABCADDCABDBACDACB is a chromosome for four teams. Here team names are A, B, C, D and One couple, say BA, signifies one gene, Team B is playing a match on Team A's ground.

2. Determine fitness of population

3. Until convergence repeat:

   a) Select parents from population
   b) Crossover and generate new population
   c) Perform mutation on new population
   d) Calculate fitness for new population

4. The algorithm comes to a termination when the population has converged. In other words, it means that the algorithm stops when a population do not produce offspring significantly different from the previous generation; there by, yielding a set of solutions to our problem using this genetic algorithm.

3.2. Particle swarm optimization

The particle swarm optimization (PSO) algorithm is a population-based stochastic optimization algorithm motivated by intelligent collective behaviour of some animals such as flocks of birds or schools of fish. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search space according to a simple mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its local best-known position but is also guided toward the best-known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. The algorithm was simplified, and it was observed to be performing optimization. PSO possesses some advantages when compared with other optimization algorithms, such as it has fewer parameters which require adjustment. It is obtained in literature that PSO has the ability to produce accurate results with consuming lesser time for computation, as well as considering lesser complexity of computation compared to some of the other bio inspired optimization techniques. Besides, parallel operation of PSO as also a good possibility. Moreover, it does not use the gradient of the problem for achieving the optimized result, i.e., differentiability of the problem is not a major issue in case of the application of PSO, which is major demand in the case of some of the other optimization schemes. Finally, a very few hyperparameters are there which are simple and do not demand for further explanation using advanced notions. PSO will work on a very wide variety of tasks on these hyperparameters. This enables PSO to become a very flexible as well as powerful algorithm.

3.2.1. Algorithm

The key steps of algorithm design are described as follows:

1. Generate particles. This is described with an example here.

   Let, BACDBDCABCADDCABDBACDACB is a chromosome for four teams. Here team names are A, B, C, D and One couple, say BA, signifies one position Team B is playing match on Team A’s ground.

2. Set initial particle best value p_{Best} and global best value g_{Best}.

3. Run a loop.

4. Run a loop for each particle.

5. Find the velocity.

6. Update the particle according to velocity.

7. If the cost of new particle is less than old particle, then change the p_{Best} for that particle.

8. If the cost of the particle is less than global best, then change the global best.

9. End the inner loop and go to no. 4

10. End.

4. Result

The results obtained (Figure 4) using both the proposed methods, i.e., using GA and PSO are described in Table 1. This table further shows a comparative analysis between the two proposed models.

Table 1: *Comparative analysis between genetic algorithm and particle swarm optimization.*

No. of teams	Cost	
	Genetic Algorithm	Particle Swarm Optimization
6	28205	21922
10	80598	60942
12	150800	116757
14	285384	205783
16	407261	292725

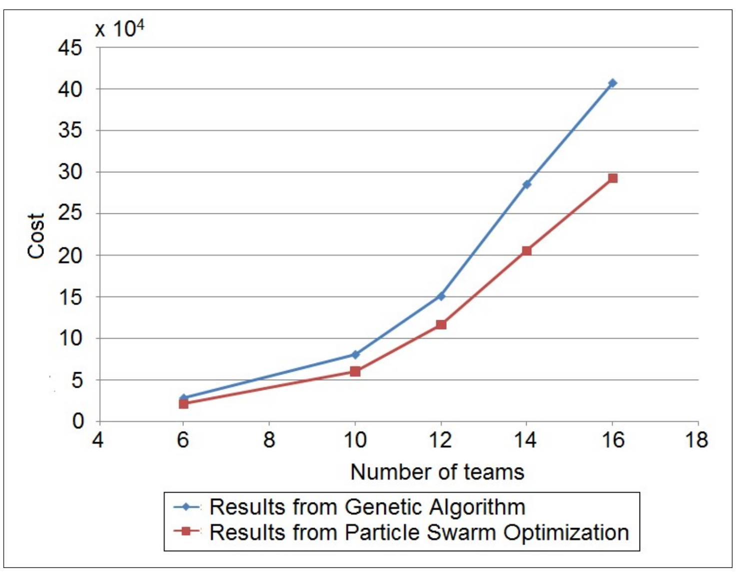

Figure 4: *Performance analysis of results obtained using Genetic Algorithm and Particle Swarm Optimization algorithms.*

We have taken NL instances from https://mat.tepper.cmu.edu/TOURN/ in order to check the efficiency of two methods. From the graphical representation we can conclude that PSO is a better option than GA in travelling tournament problem and we can see that for more teams the difference between GA and PSO is increasing significantly.

5. Conclusion

The present work discussed two algorithms genetic algorithm and particle swarm optimization for solving the travelling tournament problem. Comparing the two algorithms we conclude that particle swarm optimization will be a better choice for solving travelling tournament problem when the number of team is less than equal to 10, as the difference between their optimal results is large. In future we aim to apply these methods on the TTP with constraint and compare the computational results with other algorithms.

References

1. Thielen C, Westphal S. Complexity of the traveling tournament problem. Theoret Comput Sci. 2011;412:345-51.

2. Easton K, Nemhauser G, Trick M. Solving the travelling tournament problem: a combined integer programming and constraint programming approach. In: Burke E, De Causmaecker P. (eds), Practice and Theory of Automated Timetabling IV. PATAT 2002. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 2003.

3. Easton K, Nemhauser G, Trick M. The traveling tournament problem description and benchmarks. In: Walsh T. (eds), Principles and Practice of Constraint Programming - CP 2001. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 2001.

4. Bhattacharyya R. A note on complexity of traveling tournament problem. Optimization Online. 2009;2480.

5. Ribeiro CC, Urrutia S. Heuristics for the mirrored traveling tournament problem. Eur J Oper Res. 2007;179:775-87.

6. 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.

7. Henz M, Müller T, Thiel S. Global constraints for round robin tournament scheduling. Eur J Oper Res. 2004;153:92-101.

8. Kendall G, Knust S, Ribeiro C, et al. Scheduling in sports: an annotated bibliography. Comput Oper Res. 2010;37:1-19.

9. Rasmussen RV, Trick MA. Round robin scheduling-a survey. Eur J Oper Res. 2008;188:617-36.

10. Imahori S, Matsui T, Miyashiro R. An approximation algorithm for the unconstrained traveling tournament problem. Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, PATAT. 2010.

11. de Carvalho MAM, Lorena LAN. New models for the mirrored traveling tournament problem. Comput Ind Eng. 2012;63:1089-95.

12. Cheung KK. Solving mirrored traveling tournament problem benchmark instances with eight teams. Discrete Optim. 2008;5:138-43.

13. Biajoli FL, Lorena LAN. Clustering search approach for the traveling tournament problem. Mexican International Conference on Artificial Intelligence, Springer, Berlin, Heidelberg. 2007.

14. Urrutia S, Ribeiro CC. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. Discret Appl Math. 2006;154:1932-8.

15. Misir M, Wauters T, Verbeeck K, et al. A new learning hyper-heuristic for the traveling tournament problem. 8th Metaheuristic International Conference (MIC'09), Hamburg, Germany. 2009.

16. Costa FN, Urrutia S, Ribeiro CC. An ILS heuristic for the traveling tournament problem with predefined venues. Ann Oper Res. 2012;194:137-50.

17. Choubey NS. A novel encoding scheme for traveling tournament problem using genetic algorithm. Int J Comput Appl. 2010;2:79-82.

18. Benoist T, Laburthe F, Rottembourg B. Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems. Proceedings CPAIOR, Ashford, U.K. 2001.

19. Urrutia S, Ribeiro CC, Melo RA. A new lower bound to the traveling tournament problem. 2007 IEEE Symposium on Computational Intelligence in Scheduling, Honolulu, Hawaii. 2007.

20. Gschwind T, Irnich S. A note on symmetry reduction for circular traveling tournament problems. Eur J Oper Res. 2011;210:452-6.