A Novel Encoding Scheme for Traveling Tournament Problem using Genetic Algorithm

Dr. Nitin S. Choubey

Department of Computer Engineering,
M.P.S.T.M.E., NMIMS-deemed-to-be-University,
Shirpur Campus, Shirpur, Dhule, Maharashtra, India

ABSTRACT

Traveling Tournament Problem is a sports timetabling problem that abstracts the important issues in creating timetables where team travel is an important issue. The instances of this problem seem to be very difficult to solve even for very small cases. In this paper, Author has suggested a Novel encoding scheme for representing a solution instance. The scheme is implemented and tested for several instances of Traveling tournament problem such as NL-4, NL-6, NL-8, CIRC-4 (Constrained), CIRC-6 (Constrained), CIRC-8 (Constrained), Galaxi-4, Galaxi-6, Galaxi-8, Super-4 , Super-6 and Super-8 from Double round robin Traveling Tournament Problem. The results of the simulation are presented in the paper.

General Terms

Optimization, Scheduling, Encoding, Algorithm.

Keywords

Travelling Tournament Problem, Genetic Algorithm, Evolutionary computation, Scheduling, Encoding.

1. INTRODUCTION

The problem is originally inspired by the work done for Major League Baseball (MLB) in North America. Creating a reasonable schedule involves juggling hundreds of requests and requirements; the key issue for the schedule revolves around travel distance and flow, the pattern of home and away games in the schedule. The conflict between travel and flow is not unique to MLB. But some leagues work on Friday-Sunday schedule where teams travel from Friday-Sunday schedule. This has been explored by Campbell and Chen[1], where the was to minimize the distance traveled over such weekend pairs. Russell and Leung[2] had a similar travel objectives in their work for scheduling minor league baseball.

1.1 Traveling Tournament Overview

Given n teams with n even, a double round robin tournament is a set of games in which every team plays every other team exactly once at home and once away. A game is specified by and ordered pair of opponents. Exactly 2(n-1) slots or time periods are required to play a double round robin tournament Distance between team sites are given by an n by n distance matrix D. Each team begins at its home site and travels to play its games at the chosen venues. Each team then returns (if necessary) to its home

base at the end of the schedule [3]. Consecutive away games for a team constitute a road trip; consecutive home games are a home stand. The length of a road trip of home stand is the number of opponents played (not the travel distance).

1.1.1 *Traveling Tournament Definition*

The Traveling Tournament is defined as follows:

Input: n, the number of teams; D an n by n integer distance matrix; L, U integer parameters.

Output: A double round robin tournament [4] on the n teams such that:

- The length of every home stand and road trip is between L and U inclusive, and

- The total distance traveled by the team is minimized.

The parameters L and U define the trade-off between distance and pattern considerations. For L=1 and U=n-1, a team may take a trip equivalent to a traveling salesman tour. For small U, teams must return home often, so the distance traveled will increase.

1.1.2 *Problem Variants*

A number of researchers have developed variants of the Traveling Tournament Problem. Some of the approached which have given best solutions are:

- A Combined Integer Programming and Constraint Programming Approach [5, 6, 7],

- Non-round robin scheduling, proposed by Douglas Moody; in this problem, teams do not play a double round robin tournament but rather there is a "Match-ups" value between teams i and j, which gives the number of times i must visit j. The (regular) TTP is a Non-RR TTP with a matchup value of 1 for all i not equal to j [4].

- Relaxed TTP proposed by Renjun Bao and Michael Trick. In this variant, the schedule is not compact: teams have byes in their schedule. The number of byes is controlled by a parameter K, the number of byes per team in the schedule. K=0 corresponds to the normal Traveling Tournament Problem [4].

1.2 Genetic Algorithm Overview

The genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0's and 1's, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached (see Figure 1).
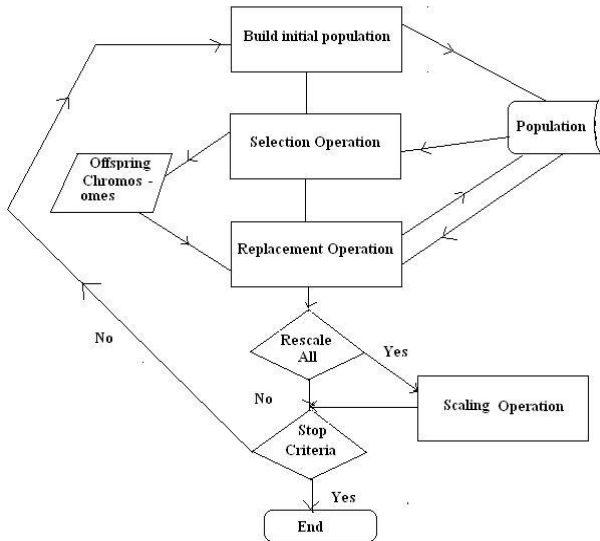

Figure 1. Simple Genetic Algorithm.

2. THE METHOD ADOPTED

In addition to the basic constraints for the TTP, the constraints for Non Mirrored-No Repeater is included in the suggested solution. The method includes a critic's module which suggests deviation of the generated solution from the objective function and the associated constraints. It suggests a penalty for every reason by which the solution is deviating from the optimum solution. This penalty is used as a parameter for calculating the fitness of the individual chromosome The symbolic chromosome is used to represent the chromosomes. The representation is biased to the number of team's participation in the tournament.

Consider the example of four teams, n=4, Here the symbols **A**, **B**, **C**, and **D** are used to represent the first, second, Third and Fourth team respectively. The random chromosome is generated based on the symbolic mapping done with the coupling done by the rearrangement of the symbols selected. One couple, say **BA**, signifies one gene, Team **B** is playing match on Team **A**'s ground.

Figure 2. The chromosome encoding scheme

The encoding scheme for the chromosome is shown in figure 2. The process of decoding a chromosome in to the solution is given below:

1. Select the chromosome & initialize **PENALTY** to 0, **Max_Cost(x,y)** to the maximum cost associated for traveling from one city to another in the entire schedule.

2. Initialize a unique number to individual match event.

3. Find the sequence of the matched to be played on each ground.

4. for each team :

   a. Calculate the **TSP** tour for each team.

   b. Increment the **PENALTY** for each **REPEATER** occurrence.

5. Calculate the Total Time (**TT**) & Traveling Cost (**TC**) for the schedule.

6. Add the difference of **TT** and the expected minimum number of slots to **PENALTY**.

7. Calculate the fitness value(**FV**) as

$$FV = TC+ PENALTY* Max\_Cost(x,y)$$

The best chromosome is the one which has zero **PENALTY** value. The example of the sample result (see Figure 2) is the **CIRCLE-4** instance [4] where **T1, T2, T3** and **T4** are represented by **A, B, C** and **D** respectively.

The Bit-Swap mutation operator and a variant of two point crossover with internal swapping [8][9] are used for the purpose of implementation. (See Figure 3).
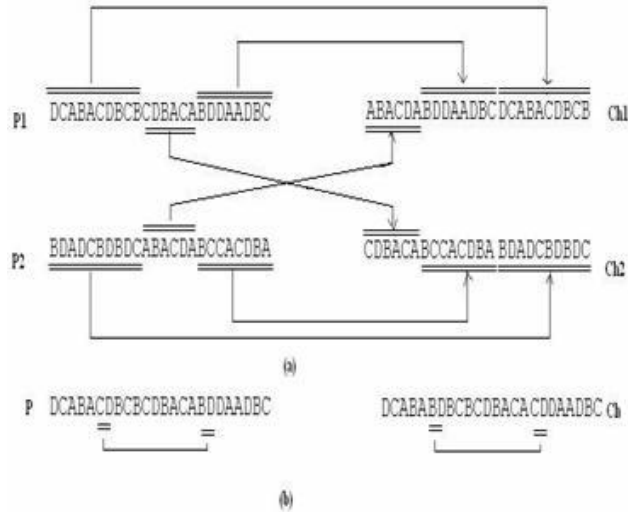

Figure 3. Crossover and Mutation operators used.

3. INSTANCE CLASSES USER AND COMPUTATIONAL RESULTS

The Benchmark instance classes[4] used for the purpose of verifying the computation results are , NL instance and circular distance instance, Super Instance and Galaxi Instance .

3.1 Circular Distance Instance

One aspect of the hardness of this problem is the embedded traveling salesman aspects. To explore this effect, "circular distance" matrices have a trivial, unique TSP solution. Number the n nodes 0 … n-1. The distance between i and j, for i>j is min {i-j, j-i+n}. The optimal tour is 0 ... n-1. The optimal TTP is further categorized as,

1. Unconstrained: No limits on length of home/away trips

2. Constrained: no more than 3 consecutive home or away

The Constrained TTP is considered for the experimentation.

3.2 NL Instance

The rules for the NL instances are as follows:

- Double round robin (A at B and B at A): n teams need 2*n-2 slots.

- No more than three consecutive home or three consecutive road games for any team

- No repeaters (A at B, followed immediately by B at A)

Objective is to minimize distance traveled (assume teams begin in their home city and must return there after the tournament). The city names for this instance are in the order ATL, NYM, PHI,

MON, FLA, PIT, CIN, CHI, STL, MIL, HOU, COL, SF, SD, LA, ARI where NL-n takes the first n cities.

3.3 Super Instances

These instances, submitted by Dave Uthus, are from the Super 14 Rugby League, composed of teams from New Zealand, Australia, and South Africa. List of team acronyms: BFN AKL CAN PRE HLM SYD JOH CHC BRI DUR DUN PER CPT WLG.

3.4 Computational Results

The example classes for the purpose of testing the methodology described in the paper are NL-4, NL-6, NL-8, CIRC-4 (Constrained), CIRC-6 (Constrained) and CIRC-8 (Constrained), Super-4, Super-6, Super-8, Galaxi-4, Galaxi-6, and Galaxi-8. The parameter U defines the tradeoff between the distance traveled and the length of Home Stand /Road trips. For the purpose of experiment U is considered as 3 i. e. any team should not play more than three consecutive rounds either away or home ground.

The results obtained are summarized in the following table (See Table 1). The population size used for the experiment is 100. Chromosome Length (CL), Total Time (TT) for the completion of the schedule and the Total cost of traveling (TCT) is given in the table.

Table 1. Computational Results

Instance	CL	TT	TCT		Optimal Solution?
			Feasible Range[4]	Actual Result	
NL-4	24	6	8276	8276	Optimal
NL-6	60	10	22969-23916	25908	Sub-Optimal
NL-8	112	14	39721-41505	41505	Sub-Optimal
CIRC-4*	24	6	20	20	Optimal
CIRC-6	60	10	64	64	Optimal
CIRC-8	112	14	132-148	146	Sub-Optimal
Galaxy-4	24	6	416	416	Optimal
Galaxy-6	60	10	1365	1365	Optimal
Galaxy-8	112	14	2373	2373	Optimal
Super-4	24	6	63405	63405	Optimal
Super-6	60	10	130635	130635	Optimal
Super-8	112	14	182409	182409	Optimal

* (See figure 2 for sample result)

It is found that the most of the instances used for the purpose of experiment have been solved by the adopted method. The instance classes for which the method has resulted in to minimum cost, as per the benchmark, is considered as optimal solution where as in the other cases it is declared as sub-optimal solution. The example of the results obtained for the instance CIRC-4 is shown in the figure 2.

4. CONCLUSION

It is found that the adapted method of introducing penalty for every deviation for the constrains and objective of the problem instance has worked successfully for the selected instances. The simulation is done for the small instances of the problem, which required to be extended for the larger instances in the benchmark. in some cases, the simulation is found to converge at the suboptimal solution, which is required to be given more attention in the future work.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] Campbell, R. T. and Chen D. S., 1976. "A Minimum Distance Basketball Scheduling Problem", in Optimal Stratgies in Sport, S.P. Ladany and R.E. Machol (eds.). North-Holland, Amsterdam.

[2] Russel, R. A. and Leung J. M., 1994. "Devising a cost effective schedule for a baseball league", Operation Research, 42,614-625.

[3] Kelly Easton, Nemhauser George L., and Trick Michael A. The Traveling Tournament Problem Description and Benchmarks. CP 2001: 580-584.

[4] Challenge Traveling Tournament Instances, August 23, 2010, from http://mat.gsia.cmu.edu/TOURN/

[5] Kelly Easton, Nemhauser George L., and Trick Michael A. Solving the Traveling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach. PATAT 2002: 100-112.

[6] Trick Michael A. Integer and Constraint Programming Approaches for Round-Robin Tournament Scheduling. PATAT 2002: 63-77.

[7] Trick Michael A. A Schedule-Then-Break Approach to Sports Timetabling. PATAT 2000: 242-253.

[8] Choubey N. S. and M. U. Kharat. "Stochastic Mutation approach for Grammar Induction using Genetic Algorithm", in the 2nd International Conference on Electronic Computer Technology (ICECT 2010), ICECT 2010,7 - 10, May 2010, Kuala Lumpur, Malaysia.

[9] Choubey, N. S. and M. U. Kharat. "Reproduction Operator Evaluation for CFG Induction using Genetic Algorithm", in Journal of Computing (Registered with the Library of Congress, USA), NY 14005-9710, USA, Journal of Computing, 2(9):89-95, September 2010, ISSN 2151-9617.