A Hybrid PSO-SA Algorithm For The Traveling Tournament Problem

Alireza Tajbakhsh[1], Kourosh Eshghi[2], Azam Shamsi[3]

[1]Faculty of Industrial Engineering, Sharif University of Technology, Tehran, Iran (atajbakhsh85@gmail.com)
[2]Faculty of Industrial Engineering, Sharif University of Technology, Tehran, Iran (eshghi@sharif.edu)
[3]Faculty of Industrial Engineering, Sharif University of Technology, Tehran, Iran (azamshamsi2004@gmail.com)

ABSTRACT

Sports scheduling has become an important area of applied operations research in recent years, since satisfying the fans and teams' requests and revenues of a sports league and TV networks may be affected by the quality of the league schedule. While this type of scheduling problem can be solved theoretically by mathematical methods, it computationally leads to hard problems. The Traveling Tournament Problem (TTP) is defined as minimizing total traveling distance for all teams in the league. In this study, a new mathematical model for the TTP with no-repeater constraint is presented. In addition, a very fast hybrid metaheuristic algorithm is proposed, which combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA). Our computational experiments on standard instances show that the hybrid approach results in comparable to or even better than current best known solutions, specifically in computational time.

***Keywords*:** Sports Scheduling, Traveling Tournament Problem, Simulated Annealing, Particle Swarm Optimization

1. INTRODUCTION

Sports scheduling has become an attractive class of scheduling problems in recent years. The combination of the optimization concept and the feasibility concept in sports scheduling problems makes them very difficult, even instances with eight teams are very sophisticated. Sports scheduling applications are characterized by constraints arising from television networks and teams.

According to [20], a *round robin tournament* is a tournament where each team plays against other teams in a predefined number of times. Most tournaments are performed as a *double round robin tournament* and each pair of teams plays twice. The games of a tournament must be assigned to some *time slots* (slots) and each team has to play no more than one game in each slot. In many leagues it is important to have an interchanging pattern of home and away games with a few breaks. A *break* is equal to two consecutive away games or two consecutive home games. A *schedule* corresponds to a timetable with allocated home and away games. When the sequences of the first and the second half of a schedule are exactly the same, it is a *mirrored* schedule. A *trip* is defined as a chain of consecutive away games, while a *home stand* presents a chain of consecutive home games.

2. LITERATURE REVIEW

There are two main classes of sports scheduling problems in the literature. The first class minimizes the number of breaks and is applied in European leagues, because each team goes back home after each away game. Schreuder [21] and de Werra [26, 27] discussed the applications of graph theory and its methods to schedule these tournaments.

The second class minimizes the total travel distance and is applied in American leagues. Campbell and Chen [4] considered the scheduling problem of a basketball league. To solve the problem, they used a 2-phase approach. Bean and Birge [2] considered a similar problem for scheduling the National Basketball Association (NBA). They constructed an Integer Programming (IP) model for the problem that was too large to solve by exact algorithms. Instead, they applied a revised version of the 2-phase method of Campbell and Chen [4]. Ferland and Fleurent [10] studied the scheduling of the National Hockey League (NHL) that was split into two conferences.

Costa [6] was the first researcher who applied a metaheuristic solution method for solving a sports scheduling problem with a distance minimization objective. He scheduled the NHL with a combination of Tabu Search (TS) and Genetic Algorithm (GA). In addition, Wright [28] presented a Simulated Annealing (SA) algorithm to schedule the National Basketball League of New Zealand.

Easton, Nemhauser and Trick [7] introduced the *Traveling Tournament Problem* (TTP) emanated from the Major League Baseball (MLB). Solutions of the TTP must satisfy difficult feasibility constraints and also minimize the total travel distances.

Many solution methods have been offered to solve the TTP. Easton et al. [7] introduced an algorithm based on a lower bound, which is the sum of the minimum travel distances for each team. Benoist et al. [3] used a combination of Lagrange Relaxation (LR) and Constraint Programming (CP). Easton et al. [8] presented a hybrid IP/CP algorithm that was a Branch and Price method. Anagnostopoulos et al. [1] developed a Simulated Annealing algorithm and they split the constraints into hard constraints and soft constraints. Lee et al. [17] in addition to building an IP model for the TTP with no-repeater constraint, introduced a Tabu Search (TS) for solving this problem. Henz [13] proposed to hybridize large neighborhood search and CP. Urrutia and Ribeiro [24] considered a specific class

978-1-4244-4136-5/09/$25.00 ©2009 IEEE

of the TTP and proved that this case corresponded to maximizing the number of breaks. Lim et al. [19] proposed a hybrid SA-Hill algorithm that combined Simulated Annealing and Hill-Climbing methods. Recently, new solution methods have been developed for TTP [5, 11, 12, and 25].

The rest of the paper is organized as follows: In section 3, TTP is described and then a new mathematical modeling is offered. In section 4, a hybrid metaheuristic method using PSO and SA is proposed. Section 5 presents the computational results and evaluates the proposed hybrid algorithm. Finally, section 6 provides concluding remarks and future research directions.

3. THE TRAVELING TOURNAMENT PROBLEM

The Traveling Tournament Problem (TTP) attempts to minimize total distance traveled by all teams. In this section, we define the main constraints of TTP, which consist of *double round robin tournament constraints*, *consecutive constraints* and *no-repeater constraints*. In addition, a new mathematical model is presented.

3.1 Problem Description

Easton et al. [7] introduced and defined TTP as follows:

✻ **Input**: *n*: the number of teams
D: a symmetric distance matrix
L, *U*: integer parameters

✻ **Output**: A double round robin tournament on the *n* teams that minimizes the total distance traveled by all teams and the number of consecutive away games and consecutive home games are between *L* and *U* inclusive.

The cost of a team in a given schedule is the total distance that it must trip, starting from its venue and finishing to its venue. Two additional constraints are important. The first is *mirroring constraint* that generates mirrored schedule and the second is *no-repeater constraint* that prevents each pair of teams from playing two games in two consecutive slots. An Integer Programming formulation for the TTP with no-repeater constraint has been presented in the literature [17]. However, we cannot use this model in our proposed hybrid algorithm since the structure of its constraints is sophisticated. Therefore, we have constructed a new, equivalent model in which the structure of the constraints is straightforward. Notice that in the proposed model, only the objective function is non-linear. The notation is as follows:

n : the number of teams

$D = \left[d_{ij} \right]$: symmetric distance matrix

$1 \le i \le n$: index for the teams

$1 \le j \le n$: index for the teams

$1 \le t \le 2n-2$: index for the slots

d_{ij} : distance between team *i*'s venue and team *j*'s venue

$x_{i,j,t} = 1 \Leftrightarrow$ team *i* plays home against team *j* in slot *t*

In what follows, we present our mathematical program:

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} \sum_{t=0}^{2n-2} U(i,t) \tag{1}$$

$$\sum_{j=1}^{n} \left(x_{i,j,t} + x_{j,i,t} \right) = 1 \quad \forall 1 \le i \le n, \forall 1 \le t \le 2n-2 \tag{2}$$

$$\sum_{t=1}^{2n-2} x_{i,j,t} = 1 \quad \forall 1 \le i \ne j \le n \tag{3}$$

$$x_{i,i,t} = 0 \quad \forall 1 \le i \le n, \forall 1 \le t \le 2n-2 \tag{4}$$

$$\begin{aligned} x_{i,j,t} + x_{j,i,t+1} \le 1 \quad \forall 1 \le i \ne j \le n, \forall 1 \le t \le 2n-3 \\ x_{j,i,t} + x_{i,j,t+1} \le 1 \quad \forall 1 \le i \ne j \le n, \forall 1 \le t \le 2n-3 \end{aligned} \tag{5}$$

$$\sum_{j=1}^{n} \sum_{s=t}^{t+3} x_{i,j,s} \le 3 \quad \forall 1 \le i \le n, \forall 1 \le t \le 2n-5 \tag{6}$$

$$\sum_{j=1}^{n} \sum_{s=t}^{t+3} x_{j,i,s} \le 3 \quad \forall 1 \le i \le n, \forall 1 \le t \le 2n-5 \tag{7}$$

$$x_{i,j,t} \in \{0,1\} \quad \forall 1 \le i,j \le n, \forall 1 \le t \le 2n-2 \tag{8}$$

Expression (1) denotes the objective function that minimizes total traveling distance for all teams in the league. Constraint (2) guarantees that each team in each slot plays against exactly one other team either at home or away. Constraint (3) means that each team must play once with any other teams in its venue. Thus, each team plays against any other team once in its home venue and once in opponent's venue. Constraint (4) prevents a team to play with itself. Constraint (5) represents the no-repeater constraint. Constraints (6) and (7) control the consecutive home games and consecutive away games, respectively. In our model, no more than three consecutive home or away games are allowed for any team with the aid of these consecutive constraints. Constraint (8) determines the type of decision variables. The total traveling distance for each team between two arbitrary slots t and $t+1$ is obtained by three constraints presented below [Constraints (9), (10), and (11)]. Notice that all teams must start in their home venues at the beginning of the league and return to their homes at the end of the league. For each $1 \le i \le n$ and each $0 \le t \le 2n-2$, the value of *U(i,t)* is equal to the distance traveled by team *i* after slot *t* and before slot $t+1$.

$$U(i,0) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} x_{i,j,1} = 1 \\ \sum_{j=1}^{n} d_{ij} x_{j,i,1} & \text{if } \sum_{j=1}^{n} x_{j,i,1} = 1 \end{cases} \tag{9}$$

$$U(i,2n-2) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} x_{i,j,2n-2} = 1 \\ \sum_{j=1}^{n} d_{ji} x_{j,i,2n-2} & \text{if } \sum_{j=1}^{n} x_{j,i,2n-2} = 1 \end{cases} \tag{10}$$

$$U(i,t)_{1 \le t \le 2n-3} = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} x_{i,j,t} = \sum_{j=1}^{n} x_{i,j,t+1} = 1 \\ \sum_{j=1}^{n} d_{ij} x_{j,i,t+1} & \text{if } \sum_{j=1}^{n} x_{i,j,t} = \sum_{j=1}^{n} x_{j,i,t+1} = 1 \\ \sum_{j=1}^{n} d_{ji} x_{j,i,t} & \text{if } \sum_{j=1}^{n} x_{j,i,t} = \sum_{j=1}^{n} x_{i,j,t+1} = 1 \\ \sum_{j=1}^{n}\sum_{j'=1}^{n} d_{jj'} x_{j,i,t} x_{j',i,t+1} & \text{if } \sum_{j=1}^{n} x_{j,i,t} = \sum_{j=1}^{n} x_{j',i,t+1} = 1 \end{cases} \quad (11)$$

3.2 Objective Function

This section describes the objective function of the TTP with no-repeater constraint. In applying the proposed metaheuristic algorithm, it is hard to satisfy all constraints of the TTP in all iterations. Anagnostopoulos et al. [1] divided them into *hard constraints* that are satisfied permanently and *soft constraints* that can be violated in an iteration. *No-repeater constraints* and *consecutive constraints* (constraints (5), (6), and (7)) indicate the soft constraints and *double round robin constraints* (constraints (2), (3), and (4)) are the hard constraints.

Since exploring the infeasible space can help to find high quality solutions for the TTP, it is necessary to modify the objective function to guide the search to feasible solutions. When for schedule S the value $\alpha = Z(S)$ is obtained from expression (1), the modified objective function for S is defined as follows [1]:

$$C(S) = \begin{cases} \alpha & S \text{ is feasible} \\ \sqrt{\alpha^2 + \left(w^* \times f\left(nbv(S)\right)\right)^2} & \text{otherwise} \end{cases} \quad (12)$$

where $nbv(S)$ presents the number of violations of the soft constraints in S and w^* shows the penalty weight. Anagnostopoulos et al. [1] showed that $f(v) = 1 +$ (sqrt(v) $\times$ Ln(v)/2) is a suitable function to control the violations of the TTP's solutions.

4. A HYBRID METAHEURISTIC ALGORITHM (PSO-SA) FOR TTP

This section describes a hybrid approach for solving the TTP with no-repeater constraint. In the proposed algorithm, two metaheuristic methods are used: Particle Swarm Optimization (PSO) and Simulated Annealing (SA). This hybrid algorithm applies a $0-1$ version of PSO in the first phase and generates many schedules rapidly. In the second phase of the hybrid algorithm, an SA approach applies the best schedules achieved in the first phase as initial schedules and improves them. The proposed algorithm leads to an optimal solution for the National League (NLn) instances of the TTP with 4, 6, and 8 teams. The information of the instances are presented in [23] and all problems have $L = 1$ and $U = 3$. This paper considers the NLn instances of the TTP with no-repeater constraint.

4.1 Particle Swarm Optimization (PSO)

PSO emanated from the behavior of bird flocking and fish schooling. Each individual attempts to keep an optimum distance from other individuals and applies its successful and effective movements which have been memorized. In this section first a brief explanation of PSO is presented and then our proposed PSO for TTP is discussed.

4.1.1 Standard PSO Algorithm

PSO applies a *population* or *swarm* of solutions to explore the search space and each *individual* or *particle* of this swarm owns a *velocity* [9, 14]. In addition, each particle has a *memory* that recalls the best position it has ever met. Suppose that the dimension of the search space and the size of the swarm are denoted by Dim and $Size$, respectively. The velocity and the position of each particle (for $r = 1,2,\ldots,Size$ and for $h = 1,2,\ldots,Dim$) is controlled by the following two equations [9, 22]:

$$v_{rh}^{iter+1} = wv_{rh}^{iter} + c_1 r_1^{iter}(p_{rh}^{iter} - x_{rh}^{iter}) + c_2 r_2^{iter}(p_{gh}^{iter} - x_{rh}^{iter}) \quad (13)$$

$$x_{rh}^{iter+1} = x_{rh}^{iter} + v_{rh}^{iter+1} \quad (14)$$

where $X_r = \left(x_{r1}, x_{r2}, \ldots, x_{rh}, \ldots, x_{r(Dim)}\right)$ is the position of the r-th particle; $P_r = \left(p_{r1}, p_{r2}, \ldots, p_{rh}, \ldots, p_{r(Dim)}\right)$ is the best previously visited position of the r-th particle; $V_r = \left(v_{r1}, v_{r2}, \ldots, v_{rh}, \ldots, v_{r(Dim)}\right)$ is the velocity of the r-th particle; g is the index of the best so far particle in the swarm; *iter* is the iteration number of the PSO; c_1, c_2 are two positive constants, called *cognitive* and *social* parameter respectively; w is called *inertia weight*; and r_1^{iter}, r_2^{iter} are random numbers, uniformly distributed in [0,1]. For controlling the velocity, we always have $-V_{\max} \le v_{rh}^{iter+1} \le V_{\max}$.

Kennedy and Eberhart [15] proposed the first discrete version of PSO for binary optimization problems. After calculating the value of the v_{rh}^{iter+1}, the value of the x_{rh}^{iter+1} will be 1 with the probability of $S(v_{rh}^{iter+1})$. Liao et al. [18] used $S(x) = 1/(1+e^{-x})$ for their proposed PSO algorithm, let $rand(\ )$ denotes a random number selected from a uniform distribution in [0,1], then:

$$rand(\ ) < S\left(v_{rh}^{iter+1}\right) \Rightarrow x_{rh}^{iter+1} = 1 \quad (15)$$

$$rand(\ ) \ge S\left(v_{rh}^{iter+1}\right) \Rightarrow x_{rh}^{iter+1} = 0 \quad (16)$$

4.1.2 Proposed PSO Algorithm for TTP

This section describes a discrete PSO approach for the TTP with no-repeater constraint based on the mathematical modeling in section 3.1. In fact, each particle of this proposed PSO represents a double round robin schedule that satisfies the hard constraints.

Generating an initial swarm of the particles is the first step for each PSO algorithm. In this paper we use the structure of the constraints (2), (3), and (4) to construct a heuristic process, which generates proper initial solutions (initial schedules). We explain this process for the TTP with n teams. Since this problem contains $2n-2$ slots, 0 or 1 must be assigned to each $n \times n \times (2n-2)$ binary variables of the proposed model in section 3.1. For $1 \le i \le n$ we define:

$Teami$: the set of slots in which the team i's opponent is not assigned yet

When 1 is assigned to $x_{i,j,t}$, it is necessary for slot t to belong to both sets *Teami* and *Teamj* simultaneously. In other words, we must have $t \in Teami \cap Teamj$. t is selected randomly from the set $Teami \cap Teamj$ with the probability of $1/\|Teami \cap Teamj\|$. If $x_{i,j,t}$ becomes 1, slot t must be eliminated from both sets *Teami* and *Teamj*. The process of generating a particle (a schedule) that satisfies the hard constraints is shown in Figure 1.

The velocity of the r-th particle in iteration $iter+1$ is updated according to the following equation:

$$v_{r,i,j,t}^{iter+1} = wv_{r,i,j,t}^{iter} + c_1 r_1^{iter}(p_{r,i,j,t}^{iter} - x_{r,i,j,t}^{iter}) + c_2 r_2^{iter}(p_{g,i,j,t}^{iter} - x_{r,i,j,t}^{iter}) \tag{17}$$

While there are various choices for c_1, c_2, and w in the literature [15, 18], the best results for solving the TTP using the proposed algorithm were obtained by $c_1 = c_2 = 0.5$ and $w = 1$ after applying it to the test problems. The scheme of the proposed PSO for solving the TTP will be as follows:

✻ Step 1. Generate a population of particles based on Figure 1.

✻ Step 2. Calculate the objective value of each particle. For $1 \le r \le Size$ let $P_r \leftarrow X_r$ and $C(P_r) \leftarrow C(X_r)$; in addition let $C(P_g) \leftarrow \min\{C(X_r); 1 \le r \le Size\}$; if min was obtained from $r = r_0$, for finding the best particle let $g \leftarrow r_0$ and $P_g \leftarrow X_{r_0}$.

✻ Step 3. Update the velocity of each particle of the current swarm according to equation (17), such that $v_{r,i,j,t}^{iter+1} \in [-V_{\max}, V_{\max}]$ is satisfied. Update the position of each particle according to equations (15) and (16); let $x_{r,i,j,t}^{iter+1} \leftarrow 1$ with the probability of $S(v_{r,i,j,t}^{iter+1})$ and $x_{r,i,j,t}^{iter+1} \leftarrow 0$ with the probability of $1 - S(v_{r,i,j,t}^{iter+1})$; if the obtained solution does not satisfy the hard constraints, modify this solution using a process similar to Figure 1.

✻ Step 4. Calculate the objective value of each particle.

✻ Step 5. For $1 \le r \le Size$, if $C(P_r) \ge C(X_r)$ is satisfied, let $P_r \leftarrow X_r$ and $C(P_r) \leftarrow C(X_r)$.

✻ Step 6. Determine the best particle of the current swarm and update g, P_g , and $C(P_g)$ if it is necessary.

✻ Step 7. If a stopping criterion is achieved, then present P_g , and $C(P_g)$; otherwise go to Step 3.

4.2 Simulated Annealing (SA)

SA is an efficient optimization algorithm that solves combinatorial optimization problems. This algorithm has motivated from the simulation of the annealing of solids and applied to a numerous number of optimization problems [16]. SA starts from an initial solution that is improved by generating local alterations. This algorithm explores both feasible and infeasible schedules. This ability is critical for the success of SA, because it is possible to find a good feasible solution in the *neighborhood* of an infeasible solution. This section describes a SA approach for the TTP with no-repeater constraint.

4.2.1 Proposed SA Algorithm for TTP

For a SA algorithm, it is important to define some neighborhood for the current solution. In the proposed hybrid algorithm, the neighborhood of a solution is a set of the schedules that are generated by moves designed by Anagnostopoulos et al. [1]:

✻ SwapHomes (t_i, t_j): This move changes two games between team i and team j.

✻ SwapSlots (s_k, s_l) : This move changes the games of slot k and slot l.

✻ SwapTeams (t_i, t_j): This move changes the pattern of team i and the pattern of team j, except at slots when they play together.

✻ PartialSwapSlots (t_i, s_k, s_l) : This move changes the games of team i at slots k and l.

✻ PartialSwapTeams (t_i, t_j, s_k) : This move changes the games of teams i and j at slot k.

If the current solution is a double round robin schedule, the new solution will also be a double round robin schedule. [1] used $T_{phase+1} = T_{phase} \times \beta$ as the cooling schedule and also applied a *strategic oscillation process* and a *reheating program* in SA approach. The strategic oscillation process has been used to change the weight w^* rationally. When Simulated Annealing leads to low temperatures, it is difficult to avoid from local optimum. In in this situation, reheating increases the temperature. For more details on the parameters of this algorithm, the reader is referred to Anagnostopoulos et al. [1].

5. COMPUTATIONAL EXPERIMENTS

The proposed hybrid algorithm is tested on a number of instances of the TTP with no-repeater constraint. As mentioned in section 4, the NLn instances are based on the information in MLB. The optimal solution for NL4, NL6, and NL8 has been reported in the literature, e.g., see [23]. Our algorithm leads to optimal schedules for these three instances rapidly and results fast and quality solutions for the other instances. Moreover, the computational time of this algorithm, compared to that of other studies, is much better for instances with $n > 8$. The results of the proposed PSO-SA algorithm and the comparison with the best solutions found in the literature (BEST) are shown in Table 1. For each instance, we implement 10 runs, reporting the best found solution (BFS) and the total running time (TRT) of the procedure in seconds. The computational experiments were performed on a Pentium 4, CPU 2.4GHz, 2GB RAM and the hybrid algorithm was coded in Microsoft Visual C++ 6.0. Table 1 also gives the results of the SA, LR-CP and SA-Hill approaches that were introduced in section 2. The detail of these methods is shown in [23].

In each of the 10 runs, the parameters are assigned according to Table 2. These values are determined based on the best solution found for different values of the parameters. The value of $\max iter$ describes the number of iterations for the PSO algorithm needed to obtain the initial solution for SA algorithm. In fact, this parameter represents a criterion to terminate the PSO algorithm in the hybrid algorithm. The size of each swarm in the PSO component is equal to 10 and the value of $V_{\max}$ is 4. In addition, the parameters of the strategic oscillation process [1], namely δ and θ, have a value equal to 1.04. Furthermore, T_0 and w_0 are equal to 400 and 4000, respectively.

The most important parameter is β. In Figure 2, we compare our solutions in terms of the computational times for NL10 and for different β values. For each β value, the corresponding instance is solved 10 times and the best solution is determined. The best solution is obtained when $\beta = 0.999$.

Table 1. Results and comparison of the best solution

Method		NL4	NL6	NL8	NL10
[1] SA	BFS	-	-	39721	59583
	TRT	-	-	1639	40269
[15] LR-CP	BFS	8276	23916	42517	68691
	TRT	1.15	86400	14400	86400
[19] SA-Hill	BFS	8276	23916	39721	59821
	TRT	1.7	821	4107	40289
[23]	BEST	8276	23916	39721	59436
PSO-SA	BFS	8276	23916	39721	65002
	TRT	0.2	30	1800	7200

Table 2. Parameter values for computational experiments

	max*iter*	β	*Reheat*	*Phases*	*Limit*
NL4	100	0.9999	10	100	50
NL6	1000	0.9999	15	1000	500
NL8	300	0.999	10	5000	500
NL10	1000	0.999	10	5000	500

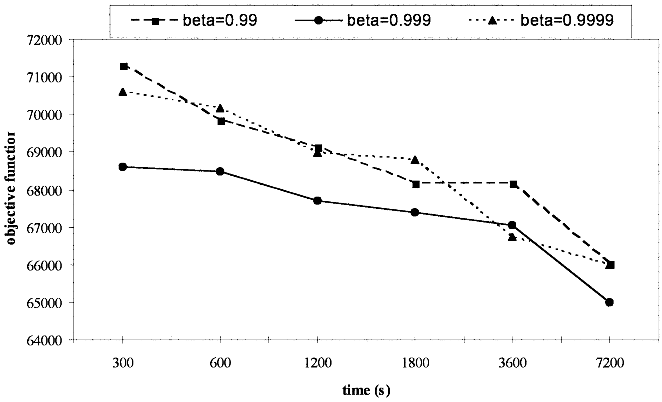

Figure 2. The effect of β on the final solution quality over time (NL10)

In addition, Figure 3 presents the performance of PSO in generating the initial solutions for the SA component of the proposed hybrid algorithm. As can be seen from this diagram, PSO is very fast and the obtained initial solutions of the NL10 instance are of high quality suitable for the SA component.

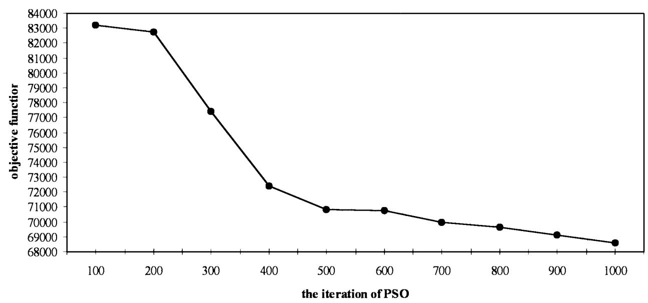

Figure 3. Analysis of initial schedules of PSO (NL10)

6. CONCLUSION

This paper studies the Traveling Tournament Problem (TTP) introduced in [7]. The TTP considers a double round robin tournament to minimize the total distance traveled by all teams. Although several recent works on the TTP have proposed heuristic approaches, only one mathematical model for the TTP exists in the literature. Our paper presents a new mathematical model using Integer Programming and suggests a hybrid metaheuristic algorithm that combines PSO and SA. The PSO component quickly generates the initial schedules, while the SA component improves these schedules and attempts to find the optimal schedule.

Our experiments show that the proposed hybrid algorithm leads to solutions that are comparable to or better than best schedules for standard instances and that it guarantees good and high quality performances.

Although a hybrid metaheuristic method is introduced in this paper, it would be attractive to investigate other hybrid metaheuristic algorithms in future research. In addition, it is important to study new effective neighborhoods to gain high quality solutions.

REFERENCES

[1] A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. "A simulated annealing approach to the traveling tournament problem", *International Workshop on Integration of AI and OR Techniques*, Montreal, 2003.

[2] J.C. Bean and J.R. Birge. "Reducing travelling costs and player fatigue in the national basketball association", *Interfaces*, 10(3), pp98–102, 1980.

[3] T. Benoist, F. Laburthe, and B. Rottembourg. "Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problems", *International Workshop on Integration of AI and OR Techniques*, Ashford, Kent UK, 2001.

[4] R.T. Campbell and D.S. Chen. "A minimum distance basketball scheduling problem", *Management Science in Sports*, Studies in the Management Sciences, 4, pp15–26, 1976.

[5] K.K.H. Cheung. "Solving mirrored traveling tournament problem benchmark instances with eight teams", *Discrete Optimization*, 5, pp138–143, 2008.

[6] D. Costa. "An evolutionary tabu search algorithm and the NHL scheduling problem", *Information Systems and Operational Research*, 33, pp161–178, 1995.

[7] K. Easton, G. Nemhauser, and M. Trick. "The traveling tournament problem: description and benchmarks", *Lecture Notes in Computer Science*, 2239, pp580–585, 2001.

[8] K. Easton, G. Nemhauser, and M. Trick. "Solving the traveling tournament problem: a combined integer programming and constraint programming approach", *Lecture Notes in Computer Science*, 2740, pp100–109, 2003.

[9] R.C. Eberhart, P. Simpson, and R. Dobbins. "Computational intelligence PC tools", *Academic Press Professional*, San Diego, CA, USA, 1996.

[10] C. Fleurent and J.A. Ferland. "Allocating games for the NHL using integer programming", *Operations Research*, 41(4), pp649–654, 1993.

[11] N. Fujiwara, S. Imahori, T. Matsui, and R. Miyashiro. "Constructive algorithms for the constant distance traveling tournament problem", *The International Series of Conferences on the Practice and Theory of Automated Timetabling*, pp402–405, 2006.

[12] L.D. Gaspero and A. Schaerf. "A composite-neighborhood tabu search approach to the traveling tournament problem", *Heuristics*, 13, pp189–207, 2007.

[13] M. Henz. "Playing with constraint programming and large neighborhood search for traveling tournaments", *The International Series of Conferences on the Practice and Theory of Automated Timetabling*, pp23–32, 2004.

[14] J. Kennedy and R.C. Eberhart. "Particle swarm optimization", *IEEE international conference on neural networks* (Perth, Australia), IEEE service center, Piscataway, pp1942–1948, 1995.

[15] J. Kennedy and R.C. Eberhart. "A discrete binary version of the particle swarm algorithm", *World multiconference on systemics*, cybernetics and informatics, Piscatawary, pp4104–4109, 1997.

[16] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. "Optimization by simulated annealing", *Science*, New Series, 220, pp671–680, 1983.

[17] J.H. Lee, Y.H. Lee, and Y.H. Lee. "Mathematical modeling and tabu search heuristic for the traveling tournament problem", *Lecture Notes in Computer Science*, 3982, pp875–884, 2006.

[18] C.-J. Liao, C.-T. Tseng, and P. Luarn. "A discrete version of particle swarm optimization for flowshop scheduling problems", *Computers & Operations Research*, 34, pp3099–3111, 2007.

[19] A. Lim, B. Rodrigues, and X. Zhang. "A simulated annealing and hill-climbing algorithm for the traveling tournament problem", *European Journal of Operational Research*, 174, pp1459–1478, 2006.

[20] R.V. Rasmussen and M.A. Trick. "Round robin scheduling – a survey", *European Journal of Operational Research*, 188, pp617–636, 2008.

[21] J.A.M. Schreuder. "Constructing timetables for sport competitions", *Mathematical Programming Study*, 13, pp58–67, 1980.

[22] Shi Y, Eberhart RC. "A modified particle swarm optimizer", *IEEE international conference on evolutionary computation*, Anchorage, Alaska, pp69–73, 1998.

[23] M.A. Trick. Michael trick's guide to sports scheduling. URL http://mat.gsia.cmu.edu/TOURN/.

[24] S. Urrutia and C.C. Ribeiro. "Minimizing travels by maximizing breaks in round robin tournament schedules", *Electronic Notes in Discrete Mathematics*, 18, pp227–233, 2004.

[25] S. Urrutia and C.C. Ribeiro. "Maximizing breaks and bounding solutions to the mirrored traveling tournament problem", *Discrete Applied Mathematics*, 154, pp1932–1938, 2006.

[26] D. de Werra. "Scheduling in sports", *Studies on Graphs and Discrete Programming*, pp381–395, 1981.

[27] D. de Werra. "Some models of graphs for scheduling sports competitions", *Discrete Applied Mathematics*, 21, pp47–65, 1988.

[28] M.B. Wright. "Scheduling fixtures for basketball New Zealand", *Computers & Operations Research*, 33, pp1875–1893, 2006.

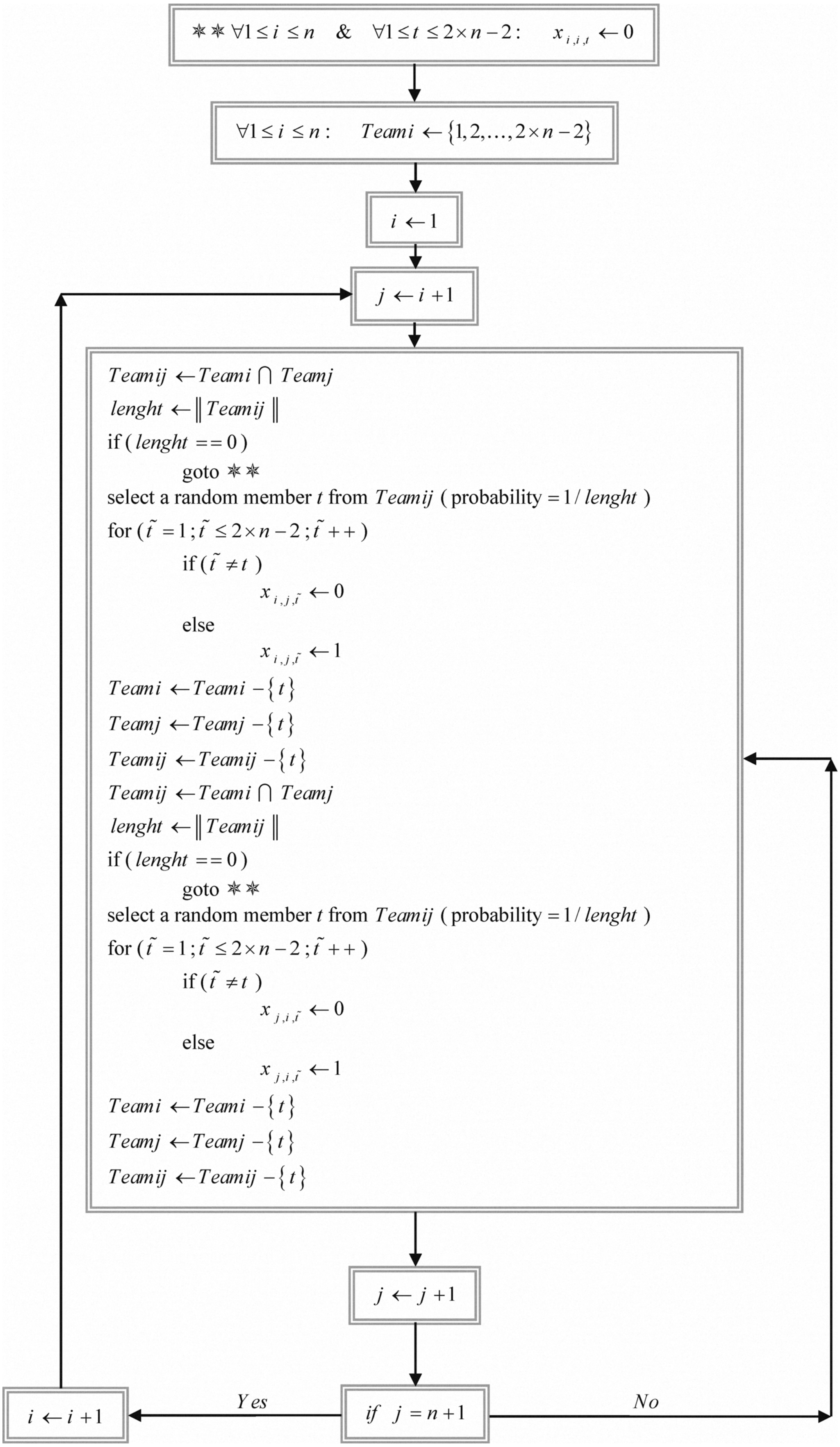

Figure 1. The heuristic process of generating an initial swarm for solving the TTP