A Hybrid Local Search Approach in Solving the Mirrored Traveling Tournament Problem

W.Wei[1,2], S.Fujimura[1], X.Wei[1], C.H.Ding[1,2]
[1] Graduate School of Information, Production and System, Waseda University, Kitakyusyu, Japan
[2] School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
(superkidphantom@gmail.com)

Abstract - **Scheduling for modern professional sports leagues has drawn considerable attention in recent years in that their practical applications involve significant revenues and generate challenging combinatorial optimization problems. The Traveling Tournament Problem is a sports scheduling problem that abstracts the important issues in creating time tables: feasibility and team travel, where the objective is to minimize the total distance traveled by the teams. In this paper, we tackle the mirrored version of this problem. First, an effective and comprehensive constructive algorithm is applied which quickly obtains initial solution at a very high quality. Then a hybrid local search approach was proposed based on the combination of Tabu Search and Variable Neighborhood Descent meta-heuristic, together with Greedy Randomized Adaptive Search Procedure, which explores large neighborhood with various and effective moves. Very competitive solutions are obtained for benchmark instances within a reasonable amount of time compared with previous results in the literature.**

Keywords - **Sports scheduling, Traveling tournament problem, Meta-heuristic, Hybrid local search, Tabu search**

I. INTRODUCTION

Professional sports leagues represent significant economic activity around the world, which involve millions of fans and huge investments in players, broadcast rights, merchandizing, and advertising and obtain tremendous amount of revenue. One key to such revenue is the schedule that teams in the league plays. Teams and leagues do not want to waste their investments in players and structure in consequence of poor schedules of games. On the other hand, sport league scheduling generates extremely challenging optimization problems with multiple constraints and objectives (involving, e.g., logistic, organizational, economical, and fairness issues).

The Traveling Tournament Problem (TTP) proposed by Easton *et al.* (2001) [1] is a typical sports scheduling problem which abstract the salient features of Major League Baseball (MLB) in the United States and was established to stimulate research in sport scheduling. It combines issues of feasibility (with regards to issues of home/away patterns) and optimality (with regards to the total distance traveled by teams). The objective is to generate a schedule which minimizes the total distance travel by the teams, subject to sophisticated feasibility constraints. Due to the difficult nature of this combination, TTP has shown to be challenging to solve for even small instances and was believed to be NP-hard. Benchmark problems are available on [2] with different sets of instances. A number of heuristic approaches (see e.g.[3], [4],[5],[6],) were applied to these instances including simulated annealing, tabu search, ant colony optimization and genetic algorithms. Yet, in most of the cases the optimal solution is still to be found.

978-1-4244-6484-5/10/$26.00 ©2010 IEEE

In this paper, we tackle the mirrored version of TTP, known as Mirrored Traveling Tournament Problem (mTTP), by using a new hybrid local search approach based on Tabu Search and Variable Neighborhood Decent (VND) together with Greedy Randomized Adaptive Search Procedure (GRASP) meta-heuristics. The paper is organized as the following: In section II, the description of TTP and mTTP is presented. In section III, we show the methodology mentioned above in detail: First a modified constructive heuristic for building good feasible initial solutions is applied. Three effective neighborhoods are described and analyzed. Then a hybrid local search procedure based on these neighborhoods with tabu search and VND is described and combined with GRASP to be the new heuristic: *HLS-mTTP*. The computational results and comparison data for 3 sets of benchmark instances are presented in section IV along with some discussion. Finally conclusion and future work is discussed in the last section.

II. PROBLEM DESCRIPTION

Given n teams with n even, a double round robin (DRR) 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. The input of TTP consists of an $n \times n$ symmetric matrix D whose d_{ij} denotes the distances between team T_i and T_j, as is shown in Table I, which is the instance generated from real-life MLB national league team air distances for 6 teams [2]. Each team begins at its home site and travels to play its games at the chosen venues. Each team then

Table I
The Distance Matrix for the NL6 Instance

	Team	*ATL*	*NYM*	*PHI*	*MON*	*FLA*	*PIT*
1	ATL	0	745	665	929	605	521
2	NYM	745	0	80	337	1090	315
3	PHI	665	80	0	380	1020	257
4	MON	929	337	380	0	1380	408
5	FLA	605	1090	1020	1380	0	1010
6	PIT	521	315	257	408	1010	0

returns (if necessary) to its home base at the end of the schedule.

For a given schedule S, the cost of a team is the total distance that it has to travel starting from its home, playing the scheduled games in S, and returning back home. The cost of a solution is defined as the sum of the cost of every team, which is the total distance travelled by all teams in the league.

The objective is to find a schedule with minimum cost, satisfying the following two constraint types:

- **C1, No repeater:** A match between two teams of T_i at T_j's home cannot be followed in the next round by the match of T_j at T_i's home.
- **C2, At most:** No more than three consecutive home or away games for any team.

The *Mirrored Traveling Tournament Problem* (mTTP) is a generalization of TTP. 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 $R = 1, 2,\dots, n-1$, with reversed venues. Repeaters will never occur in mirrored schedules. Mirrored tournaments are common for most of the soccer league schedules around the world and also for some baseball leagues.

As is shown in Fig.1, the representation is a table S indicating the opponents of the teams, where each row is a schedule corresponding to a certain team and each column is a schedule for a certain round. The opponent's representation is given by the pair (i, j), where i represents the team Ti and j represents the round R_j (e.g., the opponent of the team T_1 in round R_2 is given by $S(3,2)$, which is -2). If $S(i, j)$ is positive, the game takes place at Ti's home, otherwise at Ti's opponent home. Notice that the schedule is divided into two halves and in this work only the first half (first $n-1$ rounds) will be taken into account because as mentioned above the second half is just the mirror of first half with reversed venues and all alteration to the first half will affect the second half. The calculation of cost in Fig.1 for team 2 for example, is:

$$C_{T2} = d_{25}+ d_{52}+ d_{21}+ d_{14}+ d_{46}+ d_{62}+ d_{23}+ d_{32}$$

In previous literature, many researchers have tackled the mTTP with meta-heuristic and heuristic approaches. Ribeiro and Urrutia [7], who proposed this kind of problem, used GRASP and Iterated Local Search method to obtain very good result. After that, Biajoli and Lorena [6] tried genetic algorithm combined with SA but the result turned out to be depressing. The best known result till now for mTTP is obtained by Van Hentenryck and Vergados [8] using modified SA, but their computation costs tremendous amount of time and sometimes even takes more than 5 days for some benchmark instances.

	First half					Second half				
$T_i \backslash R_k$	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 schedule of mTTP

III. METHODOLOGY

1. *Improved Constructive Initialization*

A well initialized solution can affect greatly on the performance of meta-heuristic algorithms for timetabling problems as mentioned in [7]. In this work, we use the quick initialization inspired by them and with some modifications according to the shortage and limitation of it.

A three-stage constructive heuristic is applied here. The MDRR tournament is composed by two Single Round Robin (SRR) tournaments with the second half equal to the first except by the reversed venues. In the first stage we build a schedule for an SRR tournament using an improved SRR generating method with the concept of abstract team without assigning venues. Second the real teams are assigned to abstract teams according to statistic analysis. Finally the venues will be assigned to games between real teams and MDRR tournament is created in the third stage.

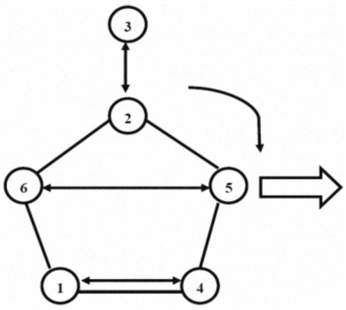
1.1 Abstract Schedule Generating

An algorithm for generating a SRR tournament known as "Polygon Method" is also used here as in [7] for n abstract teams (explained below) without assigning venues. But the difference is: Firstly, abstract teams are randomly ordered in an array as is shown in Fig. 2, rather than a simple increasing order from 1 to n.

Secondly, the traditional polygon method is executed as Fig.3: The abstract teams ordered in $1,\dots, n-1$ are initially placed at clockwise consecutively numbered nodes of a regular polygon with n - 1nodes except for the abstract team ordered n. At each round $k = 1,\dots, n-1$, each abstract team placed in the node on one side of the symmetric axis plays against its counterpart on the other side. The abstract team placed in the first node on the symmetric axis plays against the team ordered n. After each round, each abstract team ordered $1,\dots, n-1$ is moved clockwise to the immediately next node of the polygon until all $n-1$ assignments are completed. Yet in proposed method, after these procedures, the rounds will be reordered randomly rather than in the clockwise order originally.

Order	1	2	3	4	5	6
Abstract Team	2	5	4	1	6	3

Fig.2. Abstract team array for Polygon Method



R \ T	1	2	3	4	5
1	4	2	3	5	6
2	3	1	5	6	4
3	2	6	1	4	5
4	1	5	6	3	2
5	6	4	2	1	3
6	5	3	4	2	1

Fig.3. Polygon Method for $n = 6$

It is obvious that by using traditional polygon method, many team combination patterns for the schedule cannot be obtained due to its limitation. After the two improvements proposed, the limit of traditional polygon method in team patterns of the timetable is broken allowing more variety for MDRR schedule building procedures and wider search space for local search.

The abstract schedule obtained by such procedures is then duplicated and used to generate a square $n \times n$ matrix of consecutive opponents. Each entry (i, j) in this matrix is equal to the number of times the abstract teams i and j are consecutive opponents of other teams. Fig.4 displays an example of this matrix for $n = 10$, from which we can see the building procedure creates some fixed consecutive team patterns that are common in several teams' schedule (e.g. the combination of abstract team 1 and 10 appears 14 times throughout the tournament). It is likely that pairs of real teams with shorter distances should relatively be assigned to abstract teams which are consecutively played more times in the schedule to reduce total distances.

1.2 Real team assignment

According to the statistic analysis above, a quick heuristic is applied here. First, the pairs of abstract teams are sorted by non-increasing entries in the matrix of consecutive opponents. Next, pairs of teams from distance matrix D are sorted by increasing entries. We use the similar heuristics mentioned in [7] for real team assignment except that the assignment begins with assigning the first pair of real teams (t_1, t_2) in the entry to the first pair of abstract teams (α, β) in either way: t_1 to α and t_2 to β, or vice versa.

1.3 Stadium assignment

Stadium assignment will assign the venue to each

$$\begin{pmatrix} 0 & 2 & 0 & 3 & 0 & 2 & 0 & 13 & 0 & 14 \\ 2 & 0 & 0 & 4 & 0 & 13 & 0 & 2 & 13 & 0 \\ 0 & 0 & 0 & 4 & 13 & 0 & 2 & 13 & 2 & 0 \\ 3 & 4 & 4 & 0 & 3 & 4 & 4 & 4 & 4 & 4 \\ 0 & 0 & 13 & 3 & 0 & 14 & 2 & 0 & 0 & 2 \\ 2 & 13 & 0 & 4 & 14 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 4 & 2 & 0 & 0 & 0 & 13 & 13 \\ 13 & 2 & 13 & 4 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 13 & 2 & 4 & 0 & 0 & 13 & 2 & 0 & 0 \\ 14 & 0 & 0 & 4 & 2 & 1 & 13 & 0 & 0 & 0 \end{pmatrix}$$

Fig.4. Matrix of consecutive opponents for n =10

	First half					Second half				
$T_i \backslash R_k$	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

	First half					Second half				
$T_i \backslash R_k$	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.5. Schedule before (upper) and after (lower) the application of HAS

game in the schedule, in other word, to define the home-away pattern of the schedule. To minimize the total distance traveled, one should attempt to schedule as many games as possible in a road trip. Here we utilize the heuristic in [7], which can quickly generate a feasible solution in a very high quality.

2. *Neighborhood Structures*

The neighborhood of a schedule S is the set of the schedules which can be obtained by one move. Three neighborhoods are defined in this part, which compose distinct kinds of moves.

2.1 Home-Away Swap (HAS) neighborhood (S, T_i, T_j)

This move swaps the home/away roles of a game involving the teams T_i and T_j. In a SRR schedule each game of T_i versus T_j is unique. If team T_i plays home against team T_j (T_j plays away) in schedule S, then T_j plays at home and T_i plays away in schedule S' after this move. There are $n(n-1)/2$ of such moves. Fig.5 shows how this move is executed.

2.2 Team Swap (TS) neighborhood (S, T_i, T_j)

	First half					Second half				
$T_i \backslash R_k$	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

	First half					Second half				
$T_i \backslash R_k$	1	2	3	4	5	6	7	8	9	10
1	-6	4	5	-2	-3	6	-4	-5	2	3
2	5	-6	3	1	4	-5	6	-3	-1	-4
3	4	-5	-2	6	1	-4	5	2	-6	-1
4	-3	-1	6	5	-2	3	1	-6	-5	2
5	-2	3	-1	-4	-6	2	-3	1	4	6
6	1	2	-4	-3	5	-1	-2	4	3	-5

Fig.6. Schedule before (upper) and after (lower) the application of TS

	First half					Second half				
$T_i \backslash R_k$	1	2	3	4	5	6	7	8	9	10
1	-6	4		-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

	First half					Second half				
$T_i \backslash R_k$	1	2	3	4	5	6	7	8	9	10
1	-6	-5	2	4	-3	6	5	-2	-4	3
2	-5	-4	-1	3	-6	5	4	1	-3	6
3	4	6	-5	-2	1	-4	-6	5	2	-1
4	-3	2	6	-1	-5	3	-2	-6	1	5
5	2	1	3	-6	4	-2	-1	-3	6	-4
6	1	-3	-4	5	2	-1	3	4	-5	-2

Fig.7. Schedule before (upper) and after (lower) the application of RS

This move swaps the schedule of Team T_i and T_j through the tournament. To ensure that a feasible schedule is obtained, the stadium assigned to the game between T_i and T_j will also be reversed, as is represented in Fig.6. There are also $n(n-1)/2$ of such moves. Notice that schedules of other teams besides T_i and T_j should also be changed in consequence of opponent changes.

2.3 Round Swap (RS) neighborhood (S, R_k, R_l)

This move simply swaps rounds R_k and R_l, shown in Fig.7. There are $(n-1)(n-2)/2$ of such moves.

3. Hybrid Local Search Algorithm

We propose a new local search algorithm which uses the basic structure of GRASP [9] and combines tabu search [10] and VND [11] in exploring neighborhood structures of local search procedure defined in the previous part.

Fig.8 indicates the pseudo code of *HLS-mTTP*. The outer *while* loop performs the GRASP iterations, which consists of construction procedure and local search procedure. The constructive part uses the constructive algorithm mentioned in section III.1 to build a randomized initial solution S for each of the iterations.

The local search part of GRASP is a combination of VND and a partial tabu search (PTS). As is shown in Fig.8, the VND is the *repeat* loop from line 8 to 34. It numbers the neighborhoods defined in III.2 from $k = 1$ to 3 in the order of HAS, TS, then RS. For each neighborhood k, we perform the PTS which executes as what the inner *while* loop does. First all the neighbors in neighborhood k will be identified as a set $N_k(S)$ along with tabu list T_k, and then we search for the best feasible neighbor of S among all the non-tabu neighbors, namely S'_{best}. If the cost of S'_{best} is lower than S then it is accepted and local optimum will be updated in $\underline{S}$. And the move m that leads to S'_{best} will be added into T_k. When tabu list is full, the most former entry will be pushed out by the latest entry to keep the length of tabu list. If there is no feasible solution in search space, we will randomly pick one from

```
Algorithm: HLS-mTTP
1   iterator ← 1;
2   while iterator ≤ maxIter do
3       S ← constructiveInitialization();
4       iterationMinimum ← cost(S);
5       localMinimum ← cost(S);
6       k ← 1;
7       repeat
8           while termination criterion is not met do
9               Identify N_k(S);(Neighborhood Set)
10              Identify T_k;(Tabu List)
11              F = { S' | S'∈(N_k(S) − T_k) & S' is feasible};
12              if F ≠ Ø then
13                  S'_best ← arg min_{S'∈F}(cost(S'));
14                  if cost(S'_best) < cost(S) then
15                      S ← S'_best;
16                      if cost(S'_best) < localMinimum then
17                          S̲ ← S'_best
18                          localMinimum ← cost(S'_best);
19                      end if
20                  else
21                      S ← S'_best;
22                  end if
23                  updateTabuList();
24              else
25                  Randomly choose S' from N_k(S) − T_k;
26                  S ← S';
27              end if
28          end while
29          if cost(S̲) < iterationMinimum then
30              S ← S̲;
31              k ← 1;
32              iterationMinimum ← cost(S̲);
33          else
34              k++;
35          end if
36      until k > maxNeighborhoodNumber
37      if cost(S) < globalMinimum then
38          globalMinimum ← cost(S);
39          S* ← S;
40      end if
41      iterator++;
42  end while
43  return S*
```

Fig.8. Pseudo-code of *HLS-mTTP*

them in hope of finding better feasible solution after a few moves in the infeasible solution space. The *termination criterion* is met when either the number of while loops reaches the maximum bound or there is no improvement in local optimum for too many moves.

Whenever the PTS ends, it returns the local optimum solution $\underline{S}$. Once $\underline{S}$ cost less than iteration optimum in VND procedure, the neighborhood number will go back to 1, in other words, we will execute PTS from HAS neighborhood again meanwhile update the iteration optimum. If $\underline{S}$ is no better than iteration optimum, the next neighborhood will be explored. This procedure is repeated until the optimum with respect to all three neighborhoods is found. And global optimum will be updated as S^* if better solution appears whenever GRASP iteration ends.

The combination of tabu search and VND take the advantage in that PTS focuses on exploring the solution space for a certain neighborhood comprehensively and effectively. Meanwhile the VND procedure

IV. RESULTS AND DISCUSSION

To illustrate the validity and performance of *HLS-mTTP* approach, 3 sets of benchmark problems available on [2] adapted to the mirrored form are tested:

CIRCn: artificial instances with cities placed on a circle.

NLn: based on real data of the MLB in US.

Table II
Computational Results

Instances	*Best Known(TTSA2)*	*GRILS-mTTP*	*GA-SA*	***HLS-mTTP***	*Gap (%)*
CIRC8	140	140	142	148	5.72%
CIRC10	272	276	282	288	5.88%
CIRC12	432	456	458	**456**	5.55%
CIRC14	672	714	714	**714**	6.25%
CIRC16	968	1004	1014	**1002***	3.51%
CIRC18	1306(1352)	1364	1370	**1362***	4.29%
CIRC20	1852	1882	1890	**1870***	0.97%
NL8	41928	41928	43112	42802	2.08%
NL10	63832	63832	66264	64992	1.82%
NL12	119608	120655	120981	**120655**	0.87%
NL14	199363	208086	208086	**208086**	4.37%
NL16	278305	285614	290188	286532	2.96%
BRA.24	500756	506433	511256	511083	2.06%

BRA24: based on real data of the Brazilian soccer championship.

The *n* (even) for each set stands for the dimension of the instance. The max iteration for GRASP is set as 1000, tabu list size is $n - 2$ and max tabu iteration is 1000. Table II gives the results except for instances with 4 and 6 since we can easily obtain their best solutions which are certified optimal.

The results presented demonstrate the great competitiveness of the proposed methodology. In most of the instances, the results of *HLS-mTTP* surpass the ones of GA-SA [6]. And 4 results (in bold character) are equal to the GRILS-mTTP mentioned in [7], and 3 results (bold character with star) are better than theirs. All the 7 results above are the second best in the world up to the knowledge from literature and [2]. Besides the fact that some results for small instances are not so good, the approach used only three simplest neighborhoods to have got such impressive results. It is clear that the improved constructive initialization and combined local search method have enlarged the solution space and improved the accuracy and efficiency of traditional meta-heuristic approaches.

On the other hand, compared with the best known results so far, which are almost obtained in [8] (except for CIRC18), the largest gap between them and ours is about 6%, and the smallest is below 1%. The approach is very robust dealing with different kinds of instances. Yet the longest computational time on Pentium IV 3.0 GHz with 768 Mb of RAM machine was 84907 seconds. Whereas in [8] the computational time on a similar machine reached as high as 178997.5 seconds for certain instance and the averaging computational time is far longer than that in this paper. *HLS-mTTP* got all the results in a reasonable time in comparison, due to the directionality and efficiency in initializing and searching.

V. CONCLUSION

In this paper, the benchmark problem mTTP is solved by a new meta-heuristic: *HLS-mTTP*. We first develop a constructive initialization for mTTP which quickly gets feasible initial solution at a higher quality from a broader solution space. Then 3 simple but effective neighborhoods are defined and analyzed, which are latterly utilized by the hybrid local search approach.

The proposed *HLS-mTTP* meta-heuristic explores the solution space in a comprehensive and penetrating way with the combination of tabu search and VND search. Moreover, the GRASP provides with this combination a robust foundation and an extensive room. The validity and efficiency of this approach is proved by testing the benchmark instances.

Though being promising, the proposed approach did not perform well on rather smaller instances, due to the briefness of neighborhood structures. More complicated neighborhoods should be taken into account to break the limitation of boundary of existing search space. Balancing the exploration of infeasible field and computational time is also a key to effectiveness and efficiency. Furthermore, in order to get a step closer to the real-life sports scheduling operations, more realistic situations and factors can be added in to this model for future research.

REFERENCES

[1] K. Easton, G.L. Nemhauser, M.A. Trick, The traveling tournament problem: Description and benchmarks, in: T. Walsh (Ed.), Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, vol. 2239, Springer, pp. 580–584, 2001.

[2] M. Trick. http://mat.gsia.cmu.edu/TOURN/, 2002.

[3] A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. A Simulated Annealing Approach to the Traveling Tournament Problem In CP-AI-OR'2003, Montreal, Canada, May 2003.

[4] L. Di Gaspero and A. Schaerf. A composite-neighborhood tabu search approach to the traveling tournament problem. Journal of Heuristics, 13(2), pp.189–207, April 2007.

[5] P.C. Chen, G. Kendall, and G. V. Berghe. An ant based hyper-heuristic for the travelling tournament problem. In IEEE Symposium on Computational Intelligence in Scheduling, pp. 19–26, 2007.

[6] F. Biajoli, L. Lorena. Mirrored Traveling Tournament Problem: An Evolutionary Approach. Lecture Notes in Computer Science, vol 4140, springer, pp. 208-217, 2006.

[7] C. C. Ribeiro, and S. Urrutia. Heuristics for the Mirrored Traveling Tournament Problem. Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT'04), pp.323-342, 2004.

[8] P. Van Hentenryck and Y. Vergados, Traveling Tournament Scheduling: A Systematic Evaluation of Simulated Annealling, Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems,vol 3990, Lecture Notes in Computer Science, pp.228-243.2006

[9] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6 (1995) 109–133.

[10] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

[11] P. Hansen, N Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research, 130(3), pp.449-467, 2001.