European Journal of Scientific Research
ISSN 1450-216X Vol.49 No.1 (2011), pp.39-48

http://www.eurojournals.com/ejsr.htm

A Hybrid Constraint-Based Programming Approach to Design a Sports Tournament Scheduling

Soong Cai Juan
INTI University College, Centre for Pre-University Studies (COPS)
Persiaran Perdana BBN, Putra Nilai, 71800 Nilai, Negeri Sembilan, Malaysia
E-mail: soongcj1981@yahoo.com
Tel: +606-7982339

Razamin Ramli
College of Arts & Sciences, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia
E-mail: razamin@uum.edu.my
Tel: +604-9286918

Haslinda Ibrahim
College of Arts & Sciences, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia
E-mail: linda@uum.edu.my
Tel: +604-9286927

Abstract

We investigate the problem of sports tournament scheduling as reflected in the quality of tournament schedule in University Utara Malaysia (UUM). The background of the sports tournament problems that inefficiency of the human scheduler, time-consuming task and unfairness among the athletes that need to be solved gives direction and motivation in investigating the problem of scheduling the sports tournament. Subsequently, previous work related to the problem is discussed. Thus, we present an innovative hybrid of a constraint-based algorithm and a neighbourhood search, which is an exploration into alternative and improved methodology in the problem of sports tournament scheduling with special multiple constraints. A scheduling system is then developed. As a result, fair distribution of break or rest times and game venues among the competing teams are achieved in our objectives. The sports tournament scheduling system assists and improves the sports events management through high quality schedule as compared with the current human scheduler, which consider rest period, day and time preferences and venue availability. Thus, this sophisticated algorithm provides the feasible, optimum, efficient and quick solution.

Keywords: Sports Tournament Scheduling, Constraint-Based Programming, Neighbourhood Search, University Sports Events

1. Introduction

Sports management is a multidisciplinary field that integrates the sports industry and management ([1]). It can be a difficult ([2]), complex ([3]), challenging and tedious job ([4]) these days when the interest in sports has increased greatly over the last decades. It is important to construct a schedule for a sports tournament so that it can achieve the sports event's objectives ([5]), which are to handle variety needs and requirements ([4]) as well as these sports activities need to be done within a timeframe ([1]). These objectives can be achieved through efficient scheduling, which eventually has sparked our interest to undertake the case problem of sports tournament scheduling (STS) as the focus of our study.

2. Related Studies

Previous studies related to various sports tournament scheduling problems show that the problems has been solved using several techniques. For example, Simulated Annealing technique was used by [6] and [7] as well as positive semi defined programming by [8]. Mathematical programming approaches have been utilized to solve Sports League Scheduling Problem (SLSP) by [9]. Furthermore, traveling tournament problems (TTP) have also been experimented using heuristic techniques ([10]; [11]; [12]) up to 32 teams. Then again, [13] tackled the venue allocation problem in STS by a heuristic technique. Constraint Programming (CP) is another efficient technique due to its capabilities. Thus, this technique has been widely used especially in the landscapes of sports tournament scheduling ([14]; [15]; [16]; [17]; [18]; [19]; [20] and [21]). CP technique provides the optimum solution ([22]) as well as quick and feasible solution ([14]).

In addition, hybrid techniques are promising as well ([23]) in the sports tournament scheduling. Such hybrids are the integration of CP and integer programming (IP) by [17] as well as hybrid of Tabu Search (TS) and agent based technique by [24]. Due to combined advantages, capabilities and similarity of certain problems being solved with our problem at hand, a hybrid technique is thus adopted in our study.

Subsequently, this paper presents a constraint-based programming approach with application in a sports tournament scheduling problem. In applying the proposed approach, the sports tournament scheduling problem in University Utara Malaysia (UUM) is taken as a case problem and thus, modeled as a constraint satisfaction problem (CSP). Hence, the detailed explanation of the problem, the framework of the hybrid methodology and results are presented in the following sections.

In section 3, the sports tournament scheduling problem (STSP) in UUM is described. The model for the STSP and the hybrid constraint-based programming approach are illustrated in section 4 and 5, respectively. Results are shown in the last section.

3. The Sports Tournament Scheduling Problem in UUM

The sports tournament among all the residential colleges (SUKOL) organized by UUM is an annual activity involving undergraduate students in 14 residential colleges with 18 different types of games or events. These games are such as rugby, netball, futsal, *sepak takraw*, hockey (men and women), volleyball (men and women), handball (men and women), basketball (men and women), mixed badminton, mixed tennis, mixed squash, bowling (men and women) and ping pong. The tournament normally last for three to four weeks. Scheduling all events so that all sports' constraints are fulfilled is a very tedious, difficult and complex task.

It has been identified that the scheduling problem in SUKOL is due to poor scheduling, since all SUKOL events were scheduled manually. Thus, it is also very time-consuming and tedious as agreed by [25], [7] and [26]. Furthermore, SUKOL must fulfill many constraints and this has made the scheduling notoriously difficult and complex which is in parallel with statements by [27], [3], and [2]. The constraints faced by SUKOL involve possible combinations of matchups (meeting between two teams) to be played, partial round robin and single elimination strategy (playoff). Moreover, there are additional factors that need to be considered, for instance, the rest period for the teams, the availability of venue, days preference and times preference. Other scheduling problems that have been identified in

SUKOL are unfairness of venue assignment which is also suggested by [28] and rest period as also agreed by [29], [30], and [31].

Thus, this research is an effort to overcome the weaknesses in SUKOL by using an efficient hybrid constraint-based programming technique which leads to design of an improved scheduling model for a university sports tournament.

4. The STSP as a Constraint Satisfaction Problem

The SUKOL tournament scheduling problem is modeled as constraint satisfaction problem (CSP), where a finite set of variables, a function which maps every variable to a finite domain and a finite set of constraints restricting the values that the variables can simultaneously take are declared as follows. However, the variables are only properly declared when they are assigned to a matchup (meeting between two teams). The CSP then consists of Constraint Network (CN): (T, D, C) where,

- A set of teams, $T = \{ t_1, t_2,\ldots, t_n\}$ as the set of variables.
- A set of teams' domains, $D = \{D(t_1), D(t_2), D(t_3), \ldots, D(t_n)\}$ where $D(t_i)$ is a finite set of possible values for team t_i, where t_i refers to the team and $i = \{1, 2,\ldots, n\}$. In this problem, domains are the timeslots.
- A set of constraints related to teams, $C = \{c_1, c_2, \ldots , c_k \}$.
- CN or objective: to assign pairs of teams in timeslots such that all constraints are satisfied.

Constraint 1: Every team plays exactly once with every other team in prescheduled rounds.

Constraint 2: One timeslot is one hour in duration if relating to a time-based game and one and half hour in duration if relating to a score-based game.

Constraint 3: The timeslots from 12.00 noon - 4.00 pm and 7.00 pm - 9.00 pm on Fridays should be avoided.

Constraint 4: The timeslots from 9.00 am - 4.00 pm and 7.00 pm - 9.00 pm on Saturdays should be avoided.

Constraint 5: The available timeslots on Fridays are from 9.00 am - 12.00 noon, 4.00 pm - 7.00 pm and 9.00 pm - 11.00 pm.

Constraint 6: The available timeslots on Saturdays are from 4.00 pm -7.00 pm and 9.00 pm - 11.00 pm.

Constraint 7: Each team must get at least one rest period (break) before continuing to play the next match in the prescheduled rounds, where one rest period is defined as one timeslot.

Constraint 8: No more than two teams can use one venue in the same timeslot.

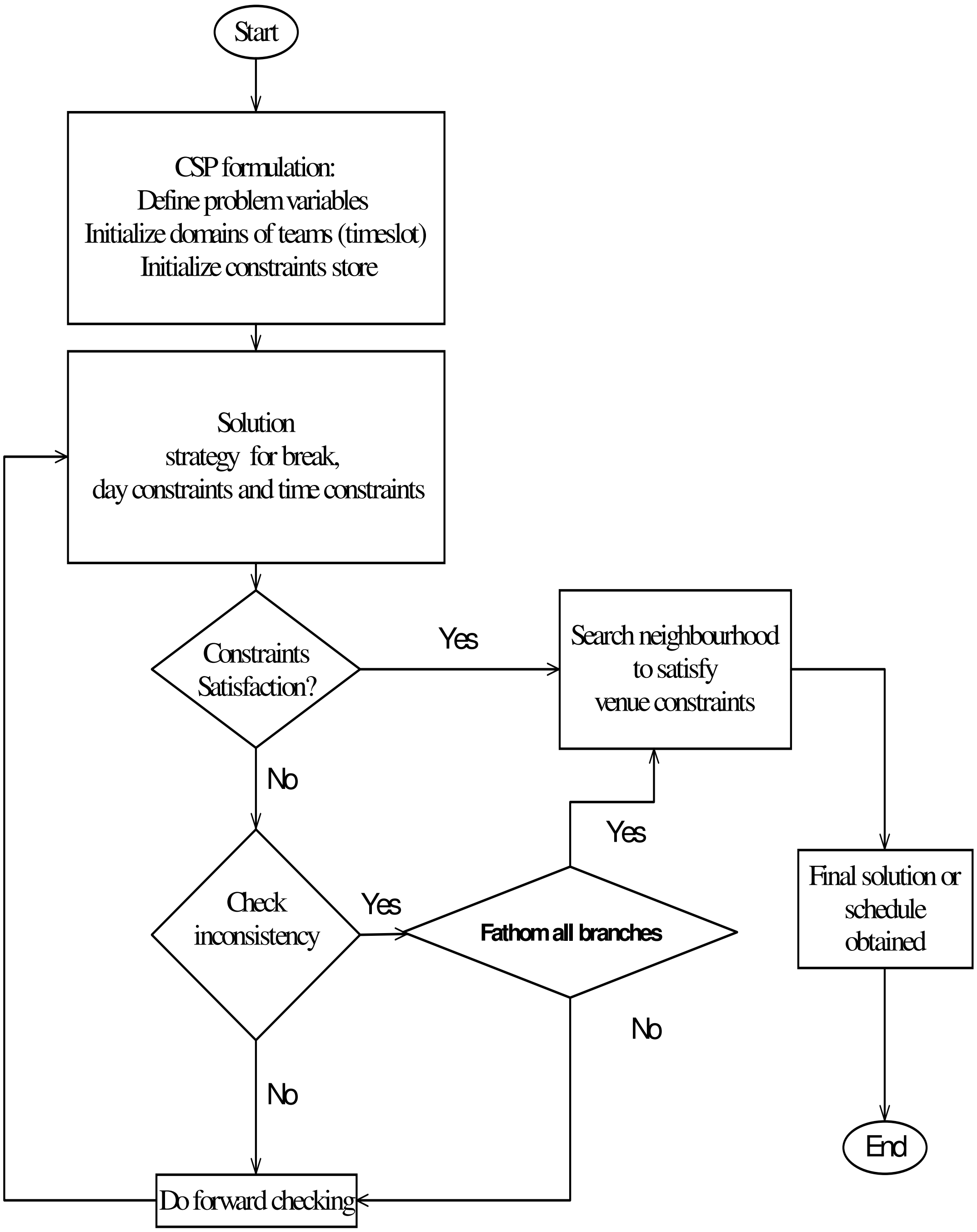
Constraint 9: Every team plays at least once in different venues during the tournament.

Constraint 10: Each team must get at least one rest period (break) before continuing to play the next match in quarter-finals, semifinals, match for third place and final (single elimination), where one rest period (break) is defined as four timeslots.

5. The Hybrid Constraint-Based Programming Approach

The proposed hybrid constraint-based programming algorithm encompasses three stages: (i) generation of possible combinations of matchups based on Partial Round Robin strategy, (ii) gathering and enumeration of all matchups and (iii) assignment of teams based on the relevant constraints through constraint-based scheduling algorithm for the preliminary round as it is the most complex stage. The approach used in this research is inspired and partly adopted from [32]. It consists of several modules such as CSP formulation, solution strategy, inconsistency check, neighbourhood search, fathom of all branches and lastly the forward checking, which are shown in Figure 1. The description of each module thus follows. In this approach, we introduce a neighbourhood search to the constraint-based programming algorithm as a hybridization strategy.

Figure 1: The architecture of the hybrid constraint-based programming approach.



5.1. Solution Strategy

The solution strategy begins by identifying timeslots (domains) and then determining matchups to allocate in the timeslots. During the allocation in a particular cell of timeslot, all matchups in the sequence list (enumerated list) would be checked one by one for break, day and time constraints satisfaction before allocating a suitable matchup in that particular cell. The process of allocating a matchup in a cell of timeslot continues until all matchups are allocated. This strategy makes formulating the constraints become more natural (where the algorithm first identifies timeslots domains), and then determines matchups to be allocated in the timeslots as suggested by [9]. At the end of the solution strategy, two possibilities arise, that are: either a solution is found (complete schedule with respect to all the three constraints) or not (compatible timeslots unable to be filled in by any matchups). If a solution is found (i.e. all matchups are filled in the schedule), then the algorithm proceeds to the next stage, that is the neighbourhood search module, after which the final solution or schedule may be found. On the other hand, if a matchup is unable to be allocated in any timeslot, then the process would proceed to the inconsistency check module.

5.2. Inconsistency Check

The function of inconsistency check module is to check and verify the dissatisfaction of all the three constraints being enforced in the solution strategy module. It detects if the particular available timeslot is not compatible with the instantiated matchups. If so, the process would proceed to the module of fathom all branches as done in [33].

5.3. Nieghbourhood Search

When a complete solution is found which means that the solution is a feasible one, then the algorithm proceeds with a neighbourhood search to satisfy venue constraint, if necessary. In this sports tournament scheduling, a neighbourhood search is referred to a swap of the two matchups in search of proper and suitable venues satisfing constraints 8 and 9. If all allocations are successfully done (final complete schedule), then the algorithm is terminated. The final solution or schedule is provided.

5.4. Fathom all Branches

If inconsistency is confirmed in the inconsistency check module, then the search tree of the problem is examined in order to check if all sub-problems or branches have been explored thoroughly (fathomed) in this module. If all branches have been fathomed, but the problem of inconsistency still exists. It means that all matchups are not able to be assigned in appropriate timeslots. So, the process would go back to the solution strategy module to search for new assignments of teams to timeslots. However, if consistency is obtained when the particular available timeslot is now compatible with an instantiated matchup as the result of feasible allocation in a branch of the problem, then a search strategy by way of a forward checking procedure is undertaken as described in the next module.

5.5. Forward Checking

In this forward checking module, a search strategy is started after the particular cell of timeslot is compatible with the instantiated matchup as an effect from the previous module. It means that the procedure of checking compatibility and feasibility would be continued to the new or next cell of the timeslot with the next instantiated matchup. However, if that new available cell of timeslot is not compatible with the new instantiated matchup, this process would be continued to check for the constraints and inconsistency repeating the checking process into the solution strategy module and so on until the find solution is obtained.

6. Results

In presenting the solution of the proposed hybrid algorithm, a prototype or a scheduling system is developed such that the natural representation of the optimal schedule is displayed. The solution (schedule) representation is in the form of 2-dimentional matrix, where column, $D(t_i)$ is the timeslot with starting and ending time and row, d_r is the day. The entry is the decision variable, t_i *vs* t_j, where t_i = team, t_j = opponent team on day, d_r and timeslot, $D(t_i)$. There are no scheduled games from Sunday to

Thursday due to classes. The value 1 in the row and column of Figure 2 shows that the particular timeslot is compatible where any matchups can be assigned in that timeslot, while 0 is vice versa. Q1 until Q4 are possible matchups assigned in feasible timeslots for teams that could proceed to the quarter phase. S1 and S2 are possible matchups assigned in feasible timeslots for teams that could play in the semi-final phase. 3rd1 is the suggested matchup timeslot for teams who fight for the third place, while F1 is the suggested matchup timeslot for teams fighting as the champion in the final phase.

Figure 2: An optimum schedule for a Netball game involving 14 teams and two venues.

Timeslot, $D(t_i)$ Day, d_k	Venue	9am-10am	10am-11am	11am-12noon	12noon-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm	8pm-9pm	9pm-10pm	10pm-11pm
Friday	V_1	team1 -vs- team4	team5 -vs- team8	team9 -vs- team11	0	0	0	0	team12 -vs- team13	team2 -vs- team4	team6 -vs- team8	0	0	team10 -vs- team11	team1 -vs- team2
	V_2	team2 -vs- team3	team6 -vs- team7	team12 -vs- team14	0	0	0	0	team1 -vs- team3	team5 -vs- team7	team9 -vs- team10	0	0	team13 -vs- team14	team3 -vs- team4
Saturday	V_1	0	0	0	0	0	0	0	team5 -vs- team6	1	1	0	0	Q1	Q3
	V_2	0	0	0	0	0	0	0	team7 -vs- team8	1	1	0	0	Q2	Q4
Friday	V_1	S1	1	1	0	0	0	0	3rd1	1	1	0	0	F1	1
	V_2	S2	1	1	0	0	0	0	1	1	1	0	0	1	1

7. Performance Evaluation

In evaluating the performance of the proposed hybrid algorithm, the scheduling result (i.e. the schedule) is evaluated in term of computational time, venue assignment and duration of break. Thus, two actual tournament schedules for academic sessions 2005/2006 and 2006/2007 are taken for comparison purposes.

7.1. Computational Time

Generating a schedule for a netball game with 14 teams playing in two available venues is considered for comparison purposes. For this game, time taken to generate the proposed hybrid tournament scheduling system is less than one minute. However, a manually generated schedule took more than a day of the scheduler's time as informed by the tournament committee. This information is based on the scheduling experience from the two actual schedules mentioned. The results are similar for the other 17 games.

7.2. Venue Assignment

Manually generated schedules in sessions 2005/2006 and 2006/2007 showed that not all teams were assigned to play in both venue 1 (i.e.V_1) and venue 2 (i.e.V_2). We generated six schedules based on the proposed hybrid algorithm. As a comparison, we highlight the venue assignment in the actual sample and the six generated schedules as exhibited in Table 1.

Venue assignments in schedules of sessions 2005/2006 and 2006/2007 show that eight teams respectively, were not assigned to play in both venue 1 and venue 2 throughout their matches. However, the six proposed schedules generated show only three teams were not assigned to play in both, venue 1 and venue 2 throughout their matches as compared to eight teams that were unable to do so in the manually generated schedules. Thus, we conclude that our proposed hybrid algorithm could generate better schedules in term of fairness in venue assignment.

Table 1: Comparison on venue assignment between the proposed algorithm and the manually generated schedules.

Schedule	2005/2006		2006/2007		Schedule 1		Schedule 2		Schedule 3		Schedule 4		Schedule 5		Schedule 6	
Venue / team	V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2
team1	3	0	3	0	1	2	1	2	1	2	2	1	2	1	2	1
team2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
team3	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
team4	1	2	1	2	3	0	3	0	3	0	2	1	2	1	2	1
team5	3	0	3	0	2	1	2	1	2	1	1	2	1	2	1	2
team6	1	2	1	2	1	2	1	2	1	2	1	2	1	2	3	0
team7	1	2	0	3	1	2	1	2	1	2	3	0	0	3	1	2
team8	1	2	1	2	2	1	2	1	2	1	1	2	2	1	1	2
team9	2	0	2	0	1	1	1	1	0	2	1	1	1	1	1	1
team10	2	0	2	0	0	2	1	1	1	1	0	2	1	1	0	2
team11	2	0	2	0	1	1	0	2	1	1	1	1	2	0	1	1
team12	0	2	0	2	1	1	1	1	1	1	1	1	1	1	1	1
team13	0	2	0	2	2	0	2	0	2	0	2	0	2	0	2	0
team14	0	2	0	2	1	1	1	1	1	1	1	1	1	1	1	1
Total number of unfairness	8		8		3		3		3		3		3		3	

7.3. Duration of Break

In computing the duration of break, we calculate the time difference between end time of a match and the start time of the next match for the same team. In this analysis, we consider 14 teams as based on the number of teams in the actual manually generated schedules. We define B_1 (as shown in Table 2) as the duration of break time for a team when playing it's first match and the second match, while B_2 (as shown in Table 2) as the duration of break time for a team when playing the second match and the third match. As we adopt Partial Round Robin strategy in generating possible matchups, there are only eight teams that play three matches while six teams only play two matches.

Table 2: Comparison on duration of break in the proposed schedules and manually generated schedules.

Schedule	2005/2006		2006/2007		Schedule 1		Schedule 2		Schedule 3		Schedule 4		Schedule 5		Schedule 6	
Time(min) / team	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2
team1	1440	2835	195	1425	360	240	360	240	360	240	360	180	360	120	360	120
team2	1440	2835	195	2745	420	180	420	180	420	180	540	1080	540	1080	540	1080
team3	1440	2835	195	2745	360	1320	360	1380	360	1320	240	60	240	60	240	60
team4	1440	2835	195	1425	420	1260	420	1260	420	1260	240	1440	240	1440	240	1440
team5	2835	1440	1785	1440	360	120	360	120	360	120	360	240	360	240	360	240
team6	2835	1440	1785	1455	540	1080	540	1080	540	1080	420	180	420	180	360	240
team7	2835	1440	1785	1455	240	60	240	60	240	60	420	240	360	300	420	180
team8	2835	1440	1785	1440	240	1440	240	1440	240	1440	360	300	420	240	480	300
team9	4275		2895		360		360		240		360		360		360	
team10	1440		1455		240		660		660		240		240		240	
team11	2835		1440		660		240		360		660		660		660	
team12	4275		2955		420		420		420		420		420		420	
team13	1440		1455		240		240		240		1260		1260		1260	
team14	2835		1455		720		720		720		1740		1020		1740	
max duration time	4275 ≈ 71.25 hours	2835 ≈ 47.25 hours	2955 ≈ 49.25 hours	2745 ≈ 45.75 hours	720 ≈ 12 hours	1440 ≈ 24 hours	720 ≈ 12 hours	1380 ≈ 23 hours	720 ≈ 12 hours	1440 ≈ 24 hours	1740 ≈ 29 hours	1440 ≈ 24 hours	1260 ≈ 21 hours	1440 ≈ 24 hours	1740 ≈ 29 hours	1440 ≈ 24 hours
min duration time	1440 ≈ 24 hours	1440 ≈ 24 hours	195 ≈ 3.25 hours	1425 ≈ 23.75 hours	240 ≈ 4 hours	60 ≈ 1 hour	240 ≈ 4 hours	60 ≈ 1 hour	240 ≈ 4 hours	60 ≈ 1 hour	240 ≈ 4 hours	60 ≈ 1 hour	240 ≈ 4 hours	60 ≈ 1 hour	240 ≈ 4 hours	60 ≈ 1 hour
Average break time per team	2442.86 ≈ 40.71 hours	2137.5 ≈ 35.63 hours	1398.21 ≈ 23.30 hours	1766.25 ≈ 29.44 hours	398.57 ≈ 6.6 hours	712.5 ≈ 11.88 hours	398.57 ≈ 6.64 hours	720 ≈ 12 hours	398.57 ≈ 6.64 hours	712.5 ≈ 11.88 hours	544.29 ≈ 9.07 hours	465 ≈ 7.75 hours	492.86 ≈ 8.21 hours	457.5 ≈ 7.63 hours	548.57 ≈ 9.14 hours	457.5 ≈ 7.63 hours

Thus, only eight teams are considered for their second duration of break time, i.e. B_2, which is reflected in Table 2. The other six teams would play two matches each thus, there are no values for their B_2. B_1 and B_2 are times in minutes. The minimum and maximum break durations, and the average break time per team are shown for each type of break durations for all schedules being examined.

The minimum break duration for the manually generated schedule is 3.25 hours, while the minimum break duration for all the proposed schedules are 1 hour, which is much less than that of the manually generated schedule. On the other hand, the maximum break duration for the manually generated schedule is 71.25 hours, while the maximum break duration for the proposed schedule is 29 hours, which is also much less that of the manually generated schedule. Subsequently, the maximum average of break duration among the manually generated schedules is 32.27 ([40.71+35.63+23.30+29.44]/4) hours, while the maximum average of break duration among the proposed schedules is 8.76 ([6.6+11.88+6.64+12+6.64+11.88+9.07+7.75+8.21+7.63+9.14+7.63]/12) hours. The proposed hybrid algorithm is able to generate better schedules than the manually generated schedules in term of short and sufficient break duration, which fulfill the problem constraint. In turn, the appropriate break durations will allow for shorter tournament timeframe, which contributes toward efficient student time management.

8. Conclusion

The proposed innovative hybrid constraint-based programming algorithm provides a feasible, efficient and quick solution as compared to the manually generated schedule in term of computational time, venue assignment and duration of break. The tournament scheduling system assists and improves the sports events management through high quality schedule, which considers rest period, day and time preferences, and venue availability. The integration of a constraint-based technique and the neighbourhood search is an exploration into alternative and improved methodology in the problem of sports tournament scheduling with special multiple constraints.

References

[1] Lussier, R. B. and Kimball, D. *Sport Management: Principles, Application, Skill Development,* South-Western: Thomson, 2004.

[2] Ribeiro, C. C. and Urrutia, S. "Heuristic for the mirrored traveling tournament problem," *European Journal of Operational Research,* vol. 179, pp.775–787, 2007.

[3] Yang, J. T., Huang, H. D., Yang, J. and J. T. Horning, "Devising a cost effective baseball scheduling by evolutionary algorithms," Taiwan, National Central University, 2002.

[4] Hassapidou, M. "Dietary assessment of five male sports teams in Greece," *Nutrition & food Science*," vol.31, no. 1, pp. 31–34, 2001.

[5] Walker, M. L. and Stotlar, D. K. *Sport Facility Management*, Boston: Jones and Bartlett, 1997.

[6] Van Hentenryck, P. and Vergados, Y. "Minimizing breaks in sport scheduling with local search," Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling, (ICAPS2005), 2005.

[7] Biajoli, F. L., Souza, M. J. F. Chaves, A. A., Mine, O. M., Cabral, L. A. F. and Pontes, R. C. "Scheduling the Brazilian Soccer Championship: A simulated annealing approach," *Proceedings of the Fifth International Conference on Practice and Theory of Automated Timetabling (PATAT2004)*, 2004.

[8] Miyashiro, R. and Matsui, T. "Semidefinite programming based approaches to the break minimization problem," *Computers and Operations Research*, 2003.

[9] McAloon, K., Trekoff, C., Wetzel, G. "Sports league scheduling," Proceedings of the 3rd ILOG Optimization Suite International Users Conference, 1997.

[10] Hamiez, J. P. and Hao, J. K. "Solving the sports league scheduling problem with tabu search," *Lecture Notes in Computer Science,* pp.24-36, 2001.

[11] Ribeiro, C. C. and Urrutia, S. "Heuristic for the Mirrored Traveling Tournament Problem," Proceedings of the Fifth International Conference on Practice and Theory of Automated Timetabling (PATAT2004), 2004.

[12] Kendall, G. Miserez, W. Vanden Berghe, G. "A constructive heuristic for the traveling tournament problem," *Proceedings of International Conference on Practice and Theory of Automated Timetabling (PATAT 2006), pp.443*-447, 2006.

[13] Burke, E. K. Werra, D. De. Silva, J. D. L. and Raess, C. "Applying heuristic methods to schedule sports competitions on multiple venues," Extended Abstract, *Proceedings of International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, pp. 441-444, 2004.

[14] Nemhauser, G. L. and Trick, M. A. "Scheduling a major college basketball conference," *Operations Research*, vol. 46, pp. 1-8, 1997.

[15] Gomes, C. P. Selman B. and Kautz, H. "Boosting combinatorial search through randomization," *Proceedings of the Fifteenth National Conference on Artificial Intelligence AAAI-98,* pp.431-437, 1998.

[16] Schaerf, A. "Scheduling sport tournaments using Constraint logic programming," *Constraints,* vol.4, pp. 43-65, 1999.

[17] Benoist, T. F. Laburthe B. and Rottembourg, "Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problem," *Proceedings of the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problem PAIOR*, pp.15-26, 2001.

[18] Regin, J. C. "Minimization of the number of breaks in sports scheduling problems using constraint programming," *Discrete Mathematics and Theoretical Computer Science*, vol.57, pp.115-130, 2001.

[19] Gan, T. L. "Constraint programming for the traveling tournament problem," Singapore, National University of Singapore, 2003.

[20] Henz, M., Muller, T., Thiel, S. "Global constraints for round robin tournament scheduling," *European Journal of Operational Research,* vol. 153, pp. 92-101, 2004.

[21] Easton, K.., Nemhauser, G. and Trick, M. "Solving the traveling tournament problem: a combined integer programming and constraint programming approach," *Proceedings of International Conference on Practice and Theory of Automated Timetabling (PATAT 2003)*, pp.100-109, 2003.

[22] Henz, M. "Playing with constraint programming and large neighborhood search for traveling tournaments," *Proceedings of International Conference on Practice and Theory of Automated Timetabling (PATAT),* 2004.

[23] Costa, D. "An evolutionary tabu search algorithm and the NHL scheduling problem," *INFOR* vol. 33, no.3, pp. 437-461, 1995.

[24] Adriaen, M., Custers N. and Vanden Berghe, G. "An agent based metaheuritic for the traveling tournament problem," Retrieved September 1, 2008 from: http://ingenieur.kahosl.be/personeel/greet.vandenberghe/artikels/SportTimetablig.pdf

[25] Glover, F. and McMillian, C. "The general employee scheduling problem: an integration of MS and AI," *Computers and Operations Research*, 13, pp. 563-573, 1986.

[26] Ribeiro, C. C. and Urrutia, S. "Scheduling the Brazilian Soccer Championship," *Proceedings of International Conference on Practice and Theory of Automated Timetabling (PATAT)*, pp.481-483, 2006.

[27] Russell, R. A. and J. Leung, M. Y. "Devising a Cost Effective Schedule For a baseball League," *Operations Research Society of America,* vol. 42, no.4, pp. 614-625, 1994.

[28] Urban, T. L. and Russell, R. A., "Scheduling Sports Competitions on Multiple Venues," *European Journal of Operational Research* vol. 148, pp. 302-311, 2003.

[29] Trick, M. A. (2004). Using sports scheduling to teach integer programming. *INFORMS Transactions on Education,* vol.5, no.1, 10-17.

[30] Miyashiro, R. and Matsui, T. "Minimizing the Carry-Over Effects in a Round-Robin Tournament," *Proceedings of International Conference on Practice and Theory of Automated Timetabling*, *PATAT,* pp.460-463, 2006.

[31] Knust, S.,"Scheduling Sports Tournament on A Single Court Minimizing Waiting Times," *Operation Research Letter*, vol. 36, pp. 471-476, 2008.

[32] Kanet, J. J., Ahire, S. L. and Gorman, M. F. "Constraint programming for Scheduling" in *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, USA: Chapman & Hall, pp. 1-21, 2004.

[33] Bartak, R. (2005). *Constraint propagation and backtracking-based search.* Retrieved September 1, 2008 from: http://www.cacs.louisiana.edu/~logan/521_f06/Doc/CPschool05notes.pdf