A Combination of PSO and Local Search in University Course Timetabling Problem

Irene, Sheau Fen Ho
Faculty of Comp. Sc. & Info. Sys.
University of Technology Malaysia
Johor Darul Ta'zim, Malaysia
ireneluv@hotmail.com

Deris, Safaai
Faculty of Comp. Sc. & Info. Sys.
University of Technology Malaysia
Johor Darul Ta'zim, Malaysia
safaai@utm.my

Mohd Hashim, Siti Zaiton
Faculty of Comp. Sc. & Info. Sys.
University of Technology Malaysia
Johor Darul Ta'zim, Malaysia
sitizaiton@utm.my

***Abstract*— The university course timetabling problem is a combinatorial optimization problem concerning the scheduling of a number of subjects into a finite number of timeslots in order to satisfy a set of specified constraints. The timetable problem can be very hard to solve, especially when attempting to find a near-optimal solutions, with a large number of instances. This paper presents a combination of particle swarm optimization and local search to effectively search the solution space in solving university course timetabling problem. Three different types of dataset range from small to large are used in validating the algorithm. The experiment results show that the combination of particle swarm optimization and local search is capable to produce feasible timetable with less computational time, comparable to other established algorithms.**

Keywords-university course timetabling problem, particle swarm optimization, local search

I. INTRODUCTION

University course timetabling problem (UCTP) is usually defined as assigning a set of events (subjects) to a number of rooms (resources) and timeslots (periods) such that a number of constraints are satisfy [2, 6]. The timetable problem can be categorized into various types (e.g. transportation timetable, examination timetable, nursing timetable etc.) by considering different specific constraints, processes and resources. The particular instance chosen for implementation is the course timetable. The course timetable problem is to organize a set of subject into a number of room and timeslot in order to satisfy a set of constraints. A combination of PSO and local search algorithm is used to schedule University Course Timetables (UCTs) and the main issues we are going to address are as follows:

- The quality of the UCT.
- The time spent in producing a timetable.

Particle swarm optimization (PSO) was originally introduced by Dr. Eberhart and Dr. Kennedy in 1995 as an optimization technique inspired by swarm intelligence and theory in general such as bird flocking, fish schooling and even human social behavior [1]. The whole idea and structure of the algorithm is inspired by evolutionary computation. Later PSO has turned out to be a worthy alternative to the standard genetic algorithm (GA) and other iterative optimization techniques [13]. PSO shares many similarities with GA [2, 3]. PSO does not use mutation and crossover operator to construct a new generation of candidate solutions like GAs does. Therefore, PSO simply modify the velocity of the particles, without the generation of a completely new population. PSO is initialized with a group of random particles and then searches for optima by updating generations. All of the particles have fitness values that evaluate by an optimized fitness function, and have velocities that update the particles flying position based on its own best experience and over the entire population. The updating policy will cause the particle swarm to move toward an area with higher objective value.

Local search is an algorithm that moves from solution to solution in the space of candidate solutions until an optimal solution is found [14]. It's starts from a candidate solution and then improves to search for a better solution in neighbor solution (feasible solution). The neighborhood search is repeated from the new solution and stop when a locally optimal solution is reached.

II. PROBLEM DESCRIPTION

In this specific timetable scheduling problem, the events are subjects, the times are the timeslots and resources are rooms. Rooms are normally limited and no two subjects should be allocated in one particular room at the same timeslot. This is the hard constraint that we are going to focus for the experimental run. In this work, there are five days for the allocation of the university courses. Each day contains nine timeslots, from eight in the morning till five in the evening. Table I illustrates the course timetable framework used for this paper, where 1,2,3,…,n are subjects, axis-x represented room 1,2,3,…,n and axis-y represented timeslots 1,2,3,…,45 [2, 4].

TABLE I. SUBJECTS ALLOCATION FOR UCTP

Time 45	Subject 1		
.			
.			
Time 3		Subject 2	
Time 2			Subject n
Time 1	Subject 3		
	Room 1	**…**	**Room k**

III. PARTICLE SWARM OPTIMIZATION

PSO can be understood by imaging a swarm of birds (particles) that search for food in an open field. Without any prior knowledge of the field, the individual birds spread out and begin their search in random locations. Each particle is

978-0-7695-3521-0/09 $25.00 © 2009 IEEE
DOI 10.1109/ICCET.2009.188

updated by following the two best values in every iteration. The first one is the best previous position of the *k*th particle at the *i*th iteration P_k^i. Another is tracked by the particle swarm optimizer and obtained so far by any particle in the population. This best position is a global best amongst all the particles G^i from the first iteration to the *i*th iteration. Each particle is equivalent to a potential solution of a problem. The velocity will adjust the particle movement, which is based on the particle's experience and experience of its neighbor. As a result, PSO can generate nearly the best solution in much lesser evolution than other algorithm [2]. The standard PSO can be described as follows.

Step 1, how many particles are used to solve the problem is decided. Every particle has its own position, velocity and best solution. Then,

$$f(P_k^i) \le f(P_k^{i-1}) \le \cdots \le f(P_k^1)$$

Step 2, the process of velocity update is shown as follows:

$$V_k^{i+1} = \chi * (w * V_k^i + c_1 * r_1 * (P_k^i - X_k^i) + c_2 * r_2 * (G^i - X_k^i))$$

Step 3, movement of the particles is processed by the following equation:

$$X_k^{i+1} = X_k^i + V_k^{i+1}$$

Where $i = 1, 2, \ldots, I$, and I is the population size. χ is a constriction factor, which is used to limit velocity and from [3, 5] we have,

$$w = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|} \text{ and } \varphi = c_1 + c_2, \varphi > 4$$

w is the inertial weight. c_1 and c_2 are two positive constants. r_1 and r_2 are two random values in the range [0, 1]. If a solution is better than G^i, G^i will be replaced by this solution to represent G^{i+1} in step 4. Otherwise, there will be no change for the global best solution, i.e. $G^i = G^{i+1}$. These recursive steps will go on until the termination criterion is met in step 5. In this case, our termination criterion will be total number of subjects.

IV. LOCAL SEARCH ALGORITHM

Local search is an iterative algorithm that moves from candidate solution S to neighbor solution S' according to some neighborhood structure [14]. Local search procedure usually consists of the following steps.

Step 1, choose an initial solution *S* to be the current solution.

Step 2, select a neighbor *S'* of the current solution *S*.

Step 3, test whether to accept the solution from *S* to *S'*. If the solution is accepted, then *S'* replaces *S* as the current solution; otherwise S is retained as the current solution.

Step 4, test whether the algorithm should terminate. If it terminates, output the best solution generated; otherwise, return to step 2.

V. COMBINATION OF PSO AND LOCAL SEARCH IN UCTP

The proposed algorithm consider of 2-stage: PSO and local search. At Each iteration, a subject would be scheduled into the timetable by applying the PSO strategies and when clashes occurred local search will be applied to seek for nearest available neighborhood timeslot and room. In the following discussion, event *s* is referred to the subject to be allocated at one particular iteration, a "best" period is the period that can be used for scheduling a subject without violating the hard constraints and with zero penalties, that is penalty = 0. An "available" period is the period that can be used for scheduling subject without violating the hard constraints and same goes for room allocation. For each event *s*, the proposed algorithm will first utilize the PSO algorithm to locate a "best" period for the subject. If clashes occurred, the second strategy: local search would be utilized to search for possible "best" period of timeslot by searching for the availability from its neighborhood timeslot and room. While assigning the subject to the timeslots and rooms, we ensure that not only the hard constraints are satisfied; the penalties incurred also must be zero. This process would be repeated until the total amount of subjects has been allocated. Figure 1 shows the particle representation for UCTP and Figure 2 shows the flowchart of the combination of PSO and Local Search in UCTP.

Subject 1	Subject 2	Subject 3		Subject n
Timeslot *j* Room *k*	Timeslot *j* Room *k*	Timeslot *j* Room *k*	………………	Timeslot *j* Room *k*

Subject=1,2,…,n, j=1,2,…,m and *k=1,2,…,p*

Figure 1. Particle representation for UCTP.

In this paper, we use a combination of PSO and Local Search algorithm to solve the discrete problem of university timetable problem. Experimental result using this algorithm into solving three different UCTP dataset will be presented in next section.

VI. EXPERIMENTAL RESULTS

The algorithm has been tested on three different dataset. The first dataset consists of 183 subjects, 21 rooms and 45 timeslots with 400 numbers of generations [6]. The second dataset consists of 74 subjects, 22 rooms and 45 timeslots with 100 numbers of generations [7]. Finally, there are 226 subjects, 16 rooms and 45 timeslots in the third dataset with 500 numbers of generations [8]. The timetable layout comprised of nine periods from Monday to Friday. This layout was repeated until the total number of subjects are fully allocate in the room and timeslot. In the PSO calculation, the particle size is set to 10 with c1=2.8, c2=1.3 [3] and w= 1 / (2 * log (2)) [9]. The experimental test were performed on a PC running Windows XP Professional SP3, with 2GB of RAM and Intel Core 2 Duo 2.2GHz CPU. The PSO implementation adopted from [10] implemented in C.

The algorithm was tested 5 times and a single clash has been given penalty of 5000 points. The work will be compared with [6, 7, 8] results; where a hybrid GA-Constraint-Based Reasoning (CBR) technique is used and also compared with standard PSO algorithm. Table II shows the result achieved by the combination of PSO and local search algorithm; while Table III shows the result achieved by the standard PSO algorithm. Lastly, Table IV shows the results achieved using hybrid GA-CBR results.

TABLE II. PERFORMANCE EVALUATION USING COMBINATION OF PSO AND LOCAL SEARCH

Performance Evaluation using Combination of PSO & Local Search						
Seed	**Data [6]**		**Data [7]**		**Data [8]**	
	Penalty	***Time (Sec.)***	***Penalty***	***Time (Sec.)***	***Penalty***	***Time (Sec.)***
1	0	19.00	0	3.00	0	28.00
2	0	19.00	0	2.00	0	28.00
3	0	18.00	0	1.00	0	28.00
4	0	18.00	0	2.00	0	28.00
5	0	18.00	0	2.00	0	28.00
Average	**0**	**18.40**	**0**	**2.00**	**0**	**28.00**

TABLE III. PERFORMANCE EVALUATION USING STANDARD PSO

Performance Evaluation using Standard PSO						
Seed	**Data [6]**		**Data [7]**		**Data [8]**	
	Penalty	***Time (Sec.)***	***Penalty***	***Time (Sec.)***	***Penalty***	***Time (Sec.)***
1	175000	18.00	10000	2.00	240000	28.00
2	190000	19.00	35000	2.00	210000	29.00
3	135000	19.00	25000	2.00	210000	28.00
4	140000	18.00	40000	2.00	260000	28.00
5	125000	18.00	15000	2.00	240000	28.00
Average	**153000**	**18.40**	**25000**	**2.00**	**232000**	**28.20**

TABLE IV. PERFORMANCE EVALUATION USING HYBRID GA-CBR

Performance Evaluation using Hybrid GA-CBR						
Seed	**Data [6]**		**Data [7]**		**Data [8]**	
	Penalty	***Time (Sec.)***	***Penalty***	***Time (Sec.)***	***Penalty***	***Time (Sec.)***
1	10000	187.00	0	18.00	0	1802.00
2	0	184.00	0	17.00	0	1832.00
3	0	182.00	0	18.00	0	5477.00
4	0	182.00	0	18.00	0	1771.00
5	0	184.00	0	18.00	0	5461.00
Average	**2000**	**183.80**	**0**	**17.80**	**0**	**3268.60**

According to the result shows in table II, we can see that combination of PSO and Local Search algorithm provide better result in solving the UCTP compare to the other two techniques; where it used less computational time and zero penalties into generating a feasible solution.

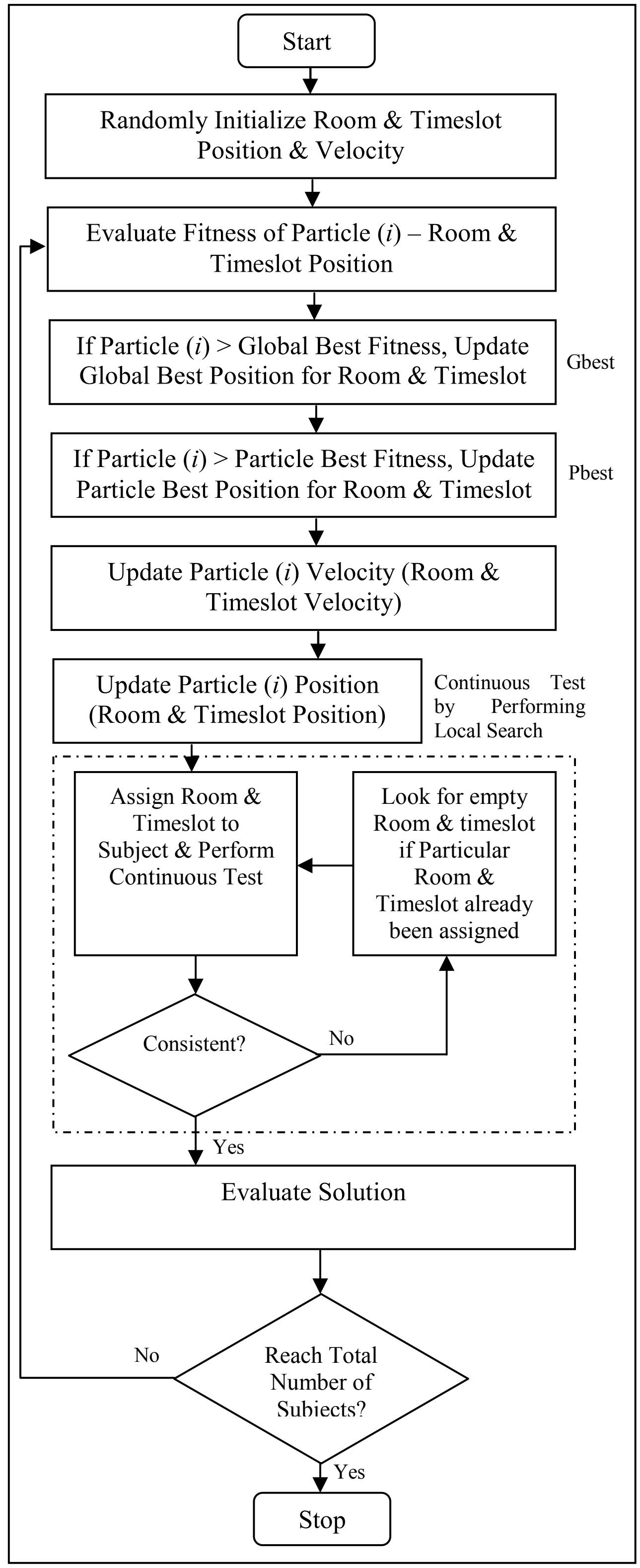

Figure 2. Flowchart of combination of PSO and Local Search in UCTP.

VII. CONCLUSION

In this paper, we have presented a model that is suitable for small to large scale of UCTP. Overall, the experiments have shown that the combination algorithm perform more efficiently and competitively on dataset under consideration. The algorithm compares well with published approaches and it is relatively easy to implement. Besides that, there are still a lot of enhancement can be made towards this algorithm into solving discrete UCTP so that it can provide better outcomes.

Subject to future research, there should be the inclusion of further constraints (i.e. soft constraints), further tuning of PSO algorithm parameters, as well as focusing on hybrid PSO with other intelligence technique to provide more feasible and better scheduling solution. To further validate the approach, different UCTP dataset will also be considered. The proposed approach could potentially be investigates with similar scheduling problem other than UCTP.

ACKNOWLEDGMENT

We would like to thank our research grant sponsor, Malaysian Ministry of Science, Technology, and Innovation (MOSTI) and University Technology of Malaysia Research Management Centre (RMC) for their support (research grant number: Vote 79240)

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in IEEE International Conference on Nueral Network, 1995, pp. 1942-1948.

[2] Shu-Chuan Chu, Yi-Tin Chen and Jiun-Huei Ho, "Timetable Scheduling Using Particle Swarm Optimization", in proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06).

[3] Fealko, R. Daniel, "Evaluating Particle Swarm Intelligence Techniques for Solving University Examination Timetabling Problems", Graduate School of computer and Information Sciences, Nova Southeastern University, 2005: A Dissertation for the degree of Doctor of Philosophy.

[4] Branimir Sigl, Marin Golub and Vedran Mornar, "Solving Timetable Scheduling Problem Using Genetic Algorithms", in the 25th International Conference Information Technology Interfaces ITI 2003, Cavtat, Croatia, June 16-19, 2003.

[5] Yin, Peng-Yeng. and Wang, Jing-Yu, "A Particle Swarm Optimization Approach to the Nonlinear Resource Allocation Problem", in Applied Mathematics and Computation, Vol. 183, 2006, pp. 232-242.

[6] Deris, Safaai., Omatu, Sigeru., Ohta, Hiroshi. and Saad, Puteh, "Incorporating Constraint Propagation in Genetic Algorithm for University Timetable Planning",in Engineering Applications of Artificial Intelligence 12, 1999, pp. 231-253.

[7] Zalmiyah, Zakaria, "Case-Based Reasoning Approach for Reactive Timetabling", Faculty of Computer Science & Information System, University of Technology Malaysia, 2001: Thesis Master.

[8] Hany, T. A. Alashwal, "The Development Of Reactive Constraint Agents For The Dynamic Timetabling Problem", Faculty of Computer Science & Information System, University of Technology Malaysia, 2003: Thesis Master

[9] [Online].Available:http://www.particleswarm.info/

[10] M. Tim Jones, AI APPLICATION PROGRAMMING Second Edition, Charles River Media, United State of America, 2005.

[11] Danial, Qarouni-Fard, Amir, Najafi-Ardabili, M-Hossein, Moeinzadeh, Sarah, Sharifian-R, Ehsan, Asgarian. and Javad, Mohammadzadeh, "Finding Feasible Timetables with Particle Swarm Optimization", in the 4th International Conference of Innovations in Information Technology, 2007, pp. 387-391.

[12] W. N. Tan, "A New Examination Timetabling Algorithm", in 16th Proceeding of National Symposium Mathematics and Science, June 3-5 2008.

[13] H. Liu, B. Li, Y. Ji and T. Sun, "Particle Swarm Optimization from lbest to gbest", Applied Soft Computing Technologies: The Challenge of Complexity, A. Abraham, B. D. Baets, M. K¨oppen, B. Nickolay (Eds.), Springer, 2006, pp. 537–545.

[14] A. Schaerf, "Local search techniques for large high-school timetabling problems", in IEEE Transactions on Systems, Man, and Cybernetics 29, volume 4, 1999, pp. 368-377.