AUTOMATED TIMETABLE GENERATOR USING PARTICLE SWARM OPTIMIZATION

Andeep Kumar
Dept. of Computer Science
Punjabi University, Patiala
Punjab, India
andeep_tayal@yahoo.com

Kawaljeet Singh
University Computer Centre
Punjabi University, Patiala
Punjab, India
singhkawaljeet@rediffmail.com

Neeraj Sharma
Dept. of Computer Science
Punjabi University, Patiala
Punjab, India
Sharma_neeraj@hotmail.com

Abstract: The timetabling problem at universities is an NP-hard problem under multiple constraints and limited resources. Thus a technique that can handle constraints is needed to optimize the problem. This paper focuses on Particle Swarm Optimization (PSO) for finding optimal solutions to the problem of course Timetabling at a Punjabi University Patiala. PSO is a promising scheme for solving NP - hard problems due to its fast convergence and fewer parameter settings. There are two objectives in this. First provide a detailed introduction to the topic of timetabling, Particle Swarm Optimization their method and their variations. The second objective is to apply them to the problem of Course Timetabling. The proposed algorithm is tested using the timetabling data from Department of Computer Science, Punjabi University, Patiala.

Keywords: Particle Swarm Optimization (PSO), university timetabling

1. Introduction: Timetabling relates all activities with regards to making a timetable. Wren in 1996 defines timetabling as the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives. As a result, a timetable specifies which people will meet at which location and at what time. A timetable must meet a number of requirements and should satisfy the desires of all people as well as possible. At universities, there are many different courses, so there i s no conflict free timetable available for every student within that given time. Therefore faculty tries to find the timetable with the least conflicts. [1]

2. Problem Description: The construction of automated course timetables for academic institutions is a very difficult problem with a lot of constraints that have to be respected and a huge search space to be explored, even if the size of the problem input is not significantly large, due to the exponential number of the possible feasible timetables. On the other hand, the problem itself does not have a widely approved definition, since different departments face different variations of it. This problem has therefore proven to be a very complex. Timetables are considered feasible provided the so-called hard constraints are respected. However, to obtain high-quality timetabling solutions, soft constraints, which impose satisfaction of a set of desirable conditions for classes and teachers, should be satisfied.

2.1 Hard Constraints [5]: Hard constraints are the constraints that physically cannot be violated; a timetable in presence of violation of such hard constraints can never be acceptable. Following are the list of hard constraints:

1. Classrooms must not be double booked.
2. Every subject must be scheduled exactly once in a day.
3. Lecturers must not be double booked.
4. A lecturer must not be booked when he/she is unavailable.
5. Some classes need to be held consecutively. For example the Labs.
6. Classrooms must be large enough to hold the class scheduled in it.

2.2 Soft constraints [5]: Some constraints are less straight forward to define. Usually, these constraints must be fulfilled as well as possible. The timetable that violates these constraints is still usable, but it is not convenient for either students or teachers. Following are the soft constraints:

1. Teachers may prefer specific time slots.
2. Teachers may prefer specific rooms.
3. Some lecturers do not wish to have classes assigned consecutively in time.
4. Most students and some lecturers do not wish to have empty periods in their timetables.
5. Classes should be distributed evenly over the week.
6. Classrooms should not be booked which are much larger than the size of the class.

2.3 The Proposed Mathematical Programming Model[6]

In order to study the computational effort involved in solving the problem of interest, the following mathematical programming model is proposed.

We define the following sets to be used in the model:

I set of all teachers

J set of all courses
K set of all subjects
L set of all days available
M set of all time periods available
C number of classrooms available per timeperiod

Finally, the following decision variables will be requiredto define the problem:
X_{ijklm}=1if teacher *I* teaches course *j* subject *k* on day *l* and at time period *m*; 0 otherwise ($i\in I$, $j\in J$, $k\in K_j$,$l\in L$, $m\in M$)
Xmi = Sum of no. of class rooms that is allocated to all teachers at particular slot m, 1 if teacher *I* teaches at timeslot m; 0 otherwise ($m\in M, i\in I$)
P_{ik}= Lies between 1 and 3, each teacher teaches at least one subject and at most three subjects($i\in I, k\in K$)
L_i=load of teacher *I* per week($i\in I$)

For our problem, the objective function reflects a preference function that needs to be maximized. It refers to the total preferences of assigning courses to the teachers. The objective function is described by the expression in equation (1):

Maximize $\sum_{i\in I}\sum_{j\in J} Pref_i(j) * Pi(j)$(1)

The following depicts some of the main constraints encountered in our timetabling problem

$\sum_{i\in I}\sum_{k\in K} X_{ijklm} \leq 1$ ($j\in J$, $l\in L$, $m\in M$) (2)

Equation (2) ensures that for a particular course, only one or zero subject conducted in every time period.

$1 \leq \sum_{k\in K} P_{ik} \leq 3$ ($i\in I$)(3)

Equation (3) represents the minimum and maximum number of subjects taught by each teacher. It is assumed that each teacher has to teach at least one course and at most three subjects.

$\sum_{j\in J}\sum_{k\in K} X_{ijklm} \leq 1$ ($i\in I$, $l\in L$, $m\in M$) (4)

Equation (4) ensures that each teacher can only teach at most one course section in a particular time period.

$\sum_{i=1}^{I} X_{mi} \leq C$ ($i\in I$, $m\in M$) (5)

Equation (5) represents the constraint that at each time period, the number of course sections taught by teachers could not be more than the number of classrooms available.

$\sum_{i=1}^{L} X_{il} \leq L_i$ ($i\in I$, $l\in L$) (6)

Equation (6) calculates the load of each teacher per week.

3. Particle swarm optimization

Kennedy and Eberhart developed Particle swarm optimization algorithm in 1995. It is a technique based on particles, each particle has velocity in search space in order to find feasible solution and adjust its position according its own previous experience and neighbors' experiences. The initial position and velocity of each particle is anomaly determined. When each particle to move a new position then it will remember its personal best (Pbest) and own information. Each particle will also exchange its information to other particles and remember its global best (Gbest). Each particle has fitness value. Fitness value of each particle's current position is compared with Pbest and Gbest. If it is better than Pbest and Gbest then update the Pbest and Gbest. Equations (1,2) as shown below are applied to update the position and velocity of each particle. Fig 3.1 show the flow chart of standard PSO.[4]

Vid=Vid + c1 * r1 * (Pid - Xid) + c2 * r2 *(Pgd - Xid) (1)

Xid=Xid+Vid (2)

In the equation

Vid is the velocity component of ith particle in the dth dimension

Xid is the position component of ith particle in the dth dimension

c1 and c2 are learning factors

Pid is the position component of the Pbest of ith particle in dth dimension

Pgd is the position component of the Gbest in dth dimension

r1 and r2 are random numbers between [0,1]

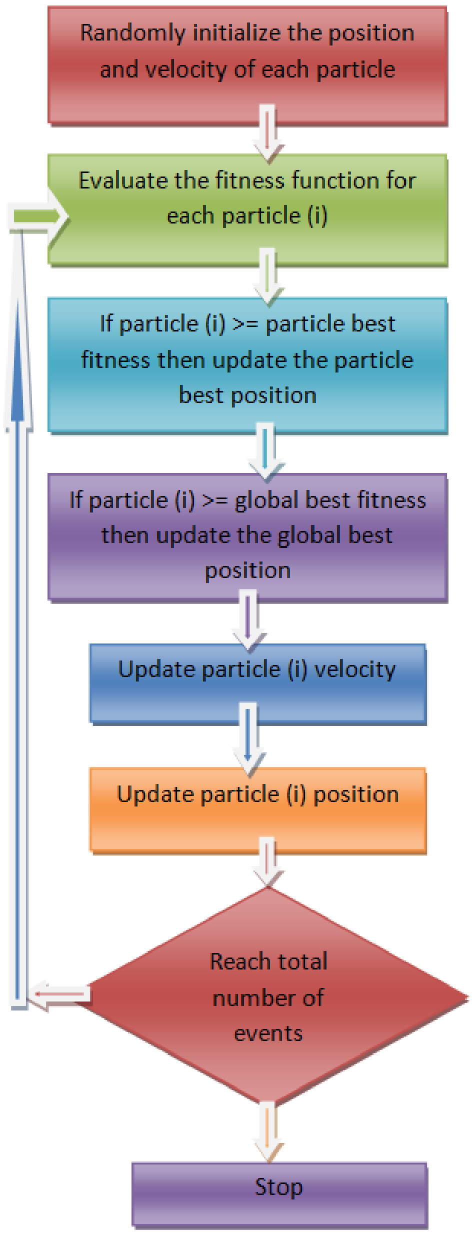


Fig 3.1Standard PSO flowchart

3.1 The Inertia Weight: Shi and Eberhart (1998) proposed the inertial weight value concept and added an inertial weight value(*w*) to the original PSO algorithm. The inertia weight value is used to balance the global search ability and the local search ability, as shown in Equation (3,4) and thereby boost the capability to locate the optimal solution and convergence rate. We use this technique in our paper.

Vid = wVid + c1 * r1 * (Pid - Xid) + c2 * r2 *(Pgd - Xid) (3)

Xid = Xid + Vid (4)

w is Inertia weight value

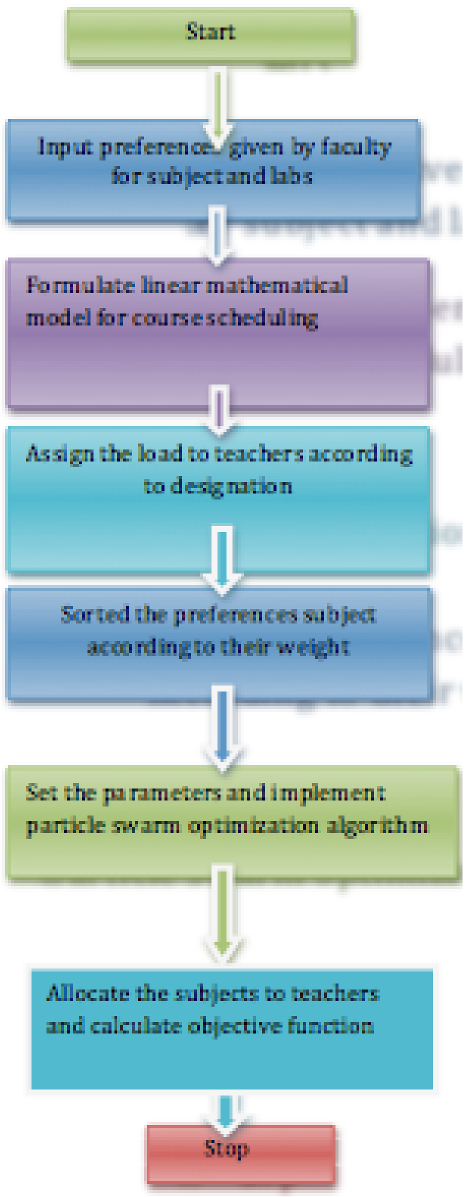
4. Department Timetabling Problem

The four main factors in course timetabling are teachers, courses, timeslots and classrooms together with other teaching facilities. The combination of these four factors is defined as the particle position and each particle represents a solution group. The objective is to obtain the optimal particle position.

First we set the four parameters of the PSO that are number of iterations, learning factors (cognitive and social) and inertia weight factor. We take the Gbest values for different number of iterations for 10, 20 ,…, 240, 250, different values of learning factors and different values of inertia weights. For which value of these parameters, Gbest is maximum that value is given optimal solution. We set these parameters and use dataset in our algorithm. We are generating the automatic timetable for Department of Computer Centre, Punjabi University Patiala. Thus we take the dataset from department and use this in our technique. [7]

The evolving process isas follows and then shows the department-timetabling flowchart.

(1) The faculty gives the preferences of subjects and software labs that they want to teach.
(2) Formulate the linear mathematical model for course scheduling on the basis of preferences.
(3) The system assigns load to faculty based on designation.
(4) The system sorted the preference of each faculty member on the basis of his or herweight.
(5) Set the parameters and implement the particle swarm algorithm for optimization results.
(6) Particle Swarm optimization allocates the subjects and software labs to faculty and calculate the objective function.



5. Results: This study was conducted by analyzing a situation involving 14 lecturers, 7 classrooms, 28 subjects, 12 software labs and 5 classes. The goal was to produce the most satisfactory class schedule to meet the various constraints as well as the expectations of the teachers. In order to demonstrate the efficiency and performance of the proposed PSO algorithm, several experiment results were carried out.

5.1 Iteration Quantity Iteration:

Figure 5.1 shows the results of analysis of the evolution of 10, 20, 30 ……., 240, 250 iterations on the given data set and find the Global best (fitness) value of course timetabling problem. When the evolution of each iteration of particles is conducted, the best Fitness value comes from when the number of iterations is 140.

5.2 Learning Factor Analysis

Using the better average satisfaction value obtained, analyzing learning factor c1 and c2, testing results with PSO are shown in figure 5.2 and 5.3.The average fitness value for c1= 0.1is higher than the other values (figure 5.2). Same as the average fitness value for c2= 1.8 is higher than others (figure 5.3).

5.3 Inertia Weight Analysis:

The inertia weight value is used to balance the global search ability and the local search ability. If the w value setting is inappropriate, it is impossible to explore unknown areas and the search for the personal best and global best will also be affected. Figure 5.4 shows the satisfaction value obtained from tests conducted with PSO using different w values. Apparently, there is higher satisfaction obtained when w= 0.9.

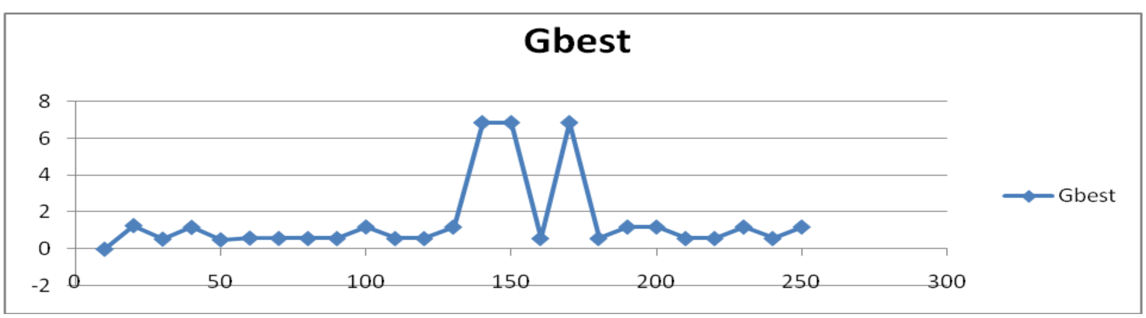


Fig 5.1Iterations verse Gbest (faculty=14,c1=0.12, c2=1.2,w=0.9)

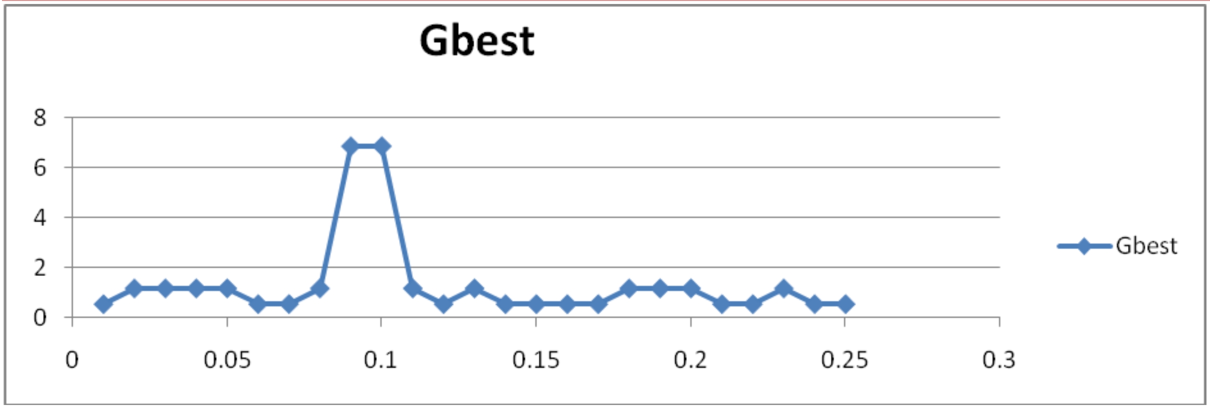


Fig 5.2 c1 verse Gbest (faculty 14, iterations 140, c2=1.2, w=0.9)

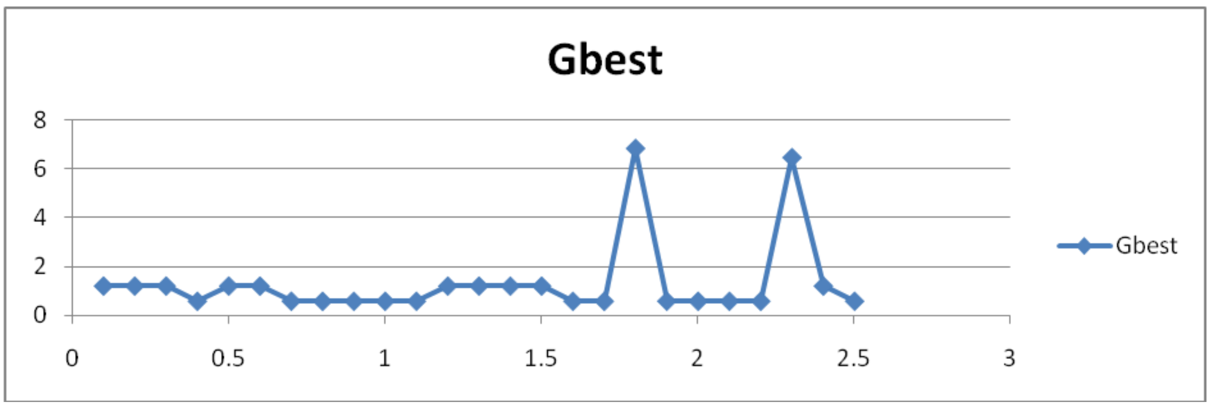


Fig 5.3c2 verse Gbest (faculty 14, iterations 140, c1=0.1,w=0.9)

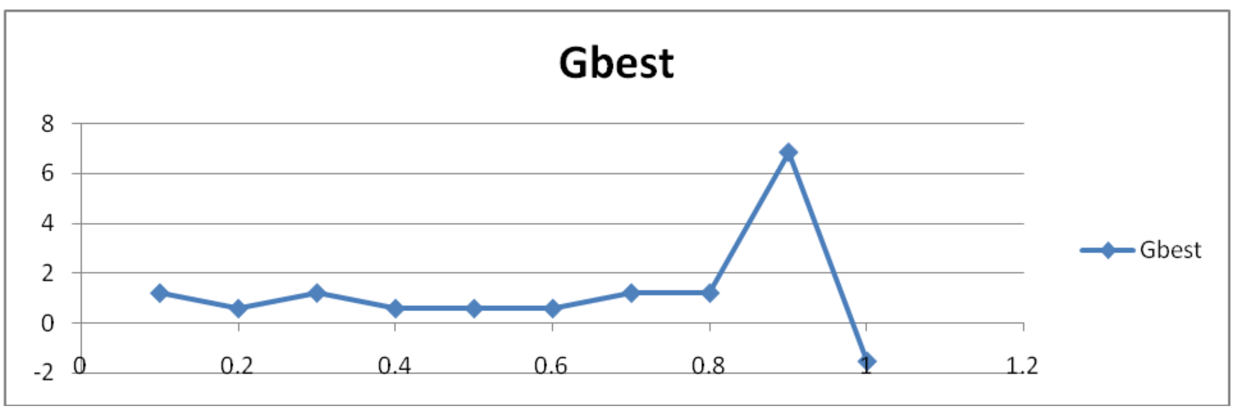


Fig 5.4w verse Gbest (faculty 14, iterations 140, c1=0.1,c2= 1.8)

Finally, course timetabling that achieves the optimal satisfaction for the dataset from department of computer science is simulated using PSO when iterations=140, c1=0.9, c2=1.8 and w=0.9. We take twenty eight subjects (S1, S2, ………., S27, S28), twelve labs (L29, L30,….., L40) and fourteen teachers (T1, T2…………. T14) in our paper. T1 to T6 are professors, T7 is associate professor and T8 to T14 are assistant professors. Each professor can teach at most one theory subject and one software lab, each associate professor can teach at most two theory subjects and one software lab and each assistant professor can teach at most three theory subjects and one software lab. Each faculty member fills their preferences for which subject they want to teach. Every faculty member fills three theory subjects and two software labs. They give a weight for every preferred subject 6 for most preferred subject, 5 for average and so onand 1 for least preferred subject. Similarly 3 for most preferred software lab, 2 for average and 1 for least preferred software labs. Table 5.1 shows the preferences for subjects given by teachers and table 5.2 shows the lab preferences and table 5.3 shows the subjects and labs that allocate to teachers.

5.4 Subject preferences by teachers

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
S1	2	0	0	0	3	0	0	0	2	0	0	2	0	3
S2	0	0	1	0	0	1	0	0	0	0	5	0	0	0
S3	0	2	0	0	0	0	5	0	0	2	0	0	0	0
S4	3	0	0	0	4	0	0	1	4	0	0	0	5	0
S5	0	0	0	1	0	0	0	0	0	0	3	0	0	0
S6	0	1	0	0	0	0	0	5	0	0	0	1	0	0
S7	0	0	0	0	0	6	0	0	0	1	0	0	0	6
S8	0	3	0	0	0	0	1	0	3	0	0	0	3	0
S9	0	0	0	0	0	0	3	0	0	0	2	0	0	4
S10	0	0	0	0	5	0	0	4	0	0	0	0	0	0
S11	0	0	0	0	0	0	0	0	0	0	0	4	0	0
S12	0	0	6	0	0	0	6	0	5	0	0	0	6	0
S13	4	0	0	2	0	0	0	0	0	0	0	0	0	5
S14	0	0	0	3	0	2	0	0	0	4	0	0	0	0
S15	0	0	2	0	0	0	0	0	0	0	1	3	0	0
S16	0	4	0	0	0	0	0	0	0	0	0	0	4	0
S17	0	0	0	0	6	0	0	6	0	0	0	0	0	0
S18	0	0	3	4	0	0	0	0	0	5	6	0	0	0
S19	5	0	0	0	0	4	0	0	0	0	0	0	0	0
S20	0	0	0	0	0	0	0	0	1	0	0	5	0	0
S21	0	0	5	0	0	0	0	0	0	0	0	0	0	0
S22	1	0	0	0	0	0	0	3	0	0	4	0	0	2
S23	0	0	0	6	0	5	4	0	0	0	0	0	0	0
S24	0	5	0	0	0	0	0	0	6	0	0	0	0	0
S25	0	0	4	0	2	0	2	0	0	6	0	6	2	0
S26	6	0	0	0	0	3	0	0	0	0	0	0	0	1
S27	0	0	0	5	0	0	0	2	0	0	0	0	1	0
S28	0	6	0	0	1	0	0	0	0	3	0	0	0	0

Table 5.1 subject preferences by teachers

5.5 Labs preferences by teachers

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
L29	1	0	0	0	0	1	0	2	0	0	0	2	0	1
L30	0	0	0	0	1	0	0	0	0	2	0	0	2	0
L31	0	1	0	3	0	0	0	0	2	0	3	0	0	0
L32	0	0	0	0	0	2	0	0	0	0	0	0	0	2
L33	2	0	1	0	0	3	0	0	1	0	0	0	3	0
L34	0	0	0	2	0	0	2	0	0	3	0	0	0	0
L35	0	2	0	0	0	0	0	1	0	0	2	3	0	0
L36	0	0	0	0	2	0	0	0	0	0	0	0	1	0
L37	3	0	3	0	0	0	3	0	0	1	0	0	0	0
L38	0	0	2	0	0	2	0	0	0	0	0	0	0	3
L39	0	3	0	0	3	0	1	3	0	0	1	0	0	0
L40	0	0	0	1	0	0	0	0	3	0	0	1	0	0

Table 5.2 lab preferences by teachers

5.6 Subject allocation to teachers

We implement this dataset in our system using particle swarm optimization and subjects and software labs are allocated to teachers as shown below:

TEACHER ID	ALLOCATED SUBJECTS AND SOFTWARE LABS ID
T1	S26,P37
T2	P39,S28
T3	S21,P38
T4	S27,P31
T5	S17,P36
T6	S7,S19,P33
T7	S3,S12,S23
T8	S6,S10,P29
T9	S24,S4,P40
T10	S25,S14,P34
T11	S22,S18,S2,P35
T12	S20,S11,S15
T13	S4,S16,S8,P30
T14	S13,S9,S1,P32

Table 5.3 subjects and software labs allocated to teachers

After the allocation of subjects to teachers system calculate the objective function. The objective function should be maximized. As much as objective function large the results would be more optimize. When we run our system the calculated objective function is 173. This value gives more satisfaction results. From our results we see linear model try to satisfy the all faculty members not only professors. We observe that junior teachers more satisfy using this technique.

6. Conclusion: PSO is promoting technique for solving complex problem such as course timetabling problem. Thus this work discusses the application of inertia weight factor a type of PSO to find solutions to solve university course timetabling. Concurrently, to reduce the computational complexity, particle encoding is designed on the basis of timeslot. The solutions found in accordance with the characteristics of the problem have been able to improve the satisfaction of the teachers and classes toward the schedule. Any conflicts between the teacher schedules, the class schedules, or the classroom schedules are also in this work.

References:

[1] Ioannis X. Tassopoulos, Grigorios N. Beligiannis, “A hybrid particle swarm optimization based algorithm for high school timetabling problems”, in applied soft computing 12(2012), 3472-3489

[2] Johor Bahru, Safaai, D. ; Zaiton, M., A Combination of PSO and Local Search in University Course Timetabling Problem, IEEE(2009),492-495

[3]Liu Su-hua, Li Lin, Study of course scheduling based on particle swarm optimization , IEEE(2011), 1692-1695

[4]Shu-Chuan Chu, Yi-Tin Chen ; Jiun-Huei Ho, Timetable Scheduling Using Particle Swarm Optimization, IEEE(2006)

[5] Ho Sheau Fen Irene, Safaai Deris, Siti Zaiton Mohd Hashim, Incorporating of Constraint-Based Reasoning into Particle Swarm Optimization for University Timetabling Problem, ISSR 1(1)2009,1-21

[6] Swarm Optimization With Local Search, Algorithms (2013), 227-24

[7] Elizabeth Montero, Maria- Cristina Riff, Leopodlo Altamirano, “A PSO Algorithm to Solve a Real Course + Exam Timetabling Problem”,ICSI (2011), 241-248.