Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 3, Issue, 8, pp.218-224, August, 2011

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

A NEW GENETIC ALGORITHM BASED POLYTECHNIC COLLEGE COURSE TIMETABLING

Dhanabalan, K[1,*]., Dr. S. Nagarajan[2] and Dr. S. Sakthivel[3]

[1]Head of Computer Science Department, SRNM College, Sattur-626203,Tamilnadu,India
[2] Prof. Department of B.Tech. (IT), Hindustan Institute of Science and Technology, Chennai, India.
[3] Principal, PSNA College of Engineering and Technology, Dindugal, Tamilnadu, India.

ARTICLE INFO

Article History:
Received 18th April, 2011
Received in revised form
26th May, 2011
Accepted 5th June, 2011
Published online 25th August, 2011

Key words:
Timetabling, Genetic algorithm, Binary weight, Crossover

ABSTRACT

The course timetabling problem is a special version of the optimization problem and it is computationally NP-hard. In this paper two methods one using binary weight and another one using normal weight instead of binary weight to teachers have been presented. Genetic algorithm in which selective two point multiple years crossover and mutation cum sequential evaluation (SMCMSE) algorithm is introduced in both the methods and test results have been compared. Both of them are proved to be useful in solving College Department as well as entire Institution Timetabling problem.

INTRODUCTION

The timetabling problem is a scheduling problem where a teacher is assigned under a timeslot in a class room of suitable capacity. This assignment depends on three different types of constraints called Physical constraints, Preference constraints and Specification constraints. These constraints depend on the nature of the Institution and its priorities. Physical constraints tells that no student or teacher can attend two different classes at the same time. Preference constraints and specification constraints says that a particular class must be held in some specified time, where specification constraints are mandatory and preference constraints are optional. Since the constraints are specific to individual problems, the development of a general technique is difficult. Among the high number of possible solutions for a scheduling problem only some of them are acceptable. The majority of the problem in scheduling is caused due to the large size and complexity of the search space. Therefore instead of using exact methods heuristic algorithms can be used to obtain near optimal solution to these kind of problems. The time tabling problem is modeled as a bi-objective problem [1] used as a basis to construct feasible assignment of teachers to classes. A binary integer programming model [2] is applied to solve school timetabling problem. The timetable problem is represented as a linear 0,1 integer programming problem [10] and the solution technique based on simplex method is used to obtain the solution. Universal method for solving large highly constrained timetabling problems from different domains is solved based on evolutionary algorithm [3] framework and operates on two levels. Tabu search algorithm [7,8] to solve class/teacher timetabling problem is also presented. Two automatic timetabling systems based on evolutionary algorithms [9] are described. Genetic algorithm based approaches [4, 5, 6, 11, 13, 14, 15, 16, 17, 18] were used to obtain optimal solution for the timetabling problem. A solution method [12] consists of two phases for solving the timetabling problem using local search methods and equipped with an interactive user–intervention facility is presented.

In this paper use of genetic algorithm for developing a common time table for a Polytechnic College is considered. The teachers are represented by binary 2^i where i is the number of the teacher. If there are n teachers the teachers are represented as 2^1, 2^2, 2^3 ……2^n. The solution found at each stage will be a feasible solution. The paper is organized as follows. In Section 2 timetabling problem representations along with the terminology and the set of constraints used are given. In section 3 the proposed methods are explained in detail. Section 4 is devoted to discuss the experimental results obtained using the proposed method.

The Timetabling Problem

The terminology and definition of the timetabling problem are described in this section.

Terminology

Student Group

Group of students admitted in a particular major during the same academic year

***Corresponding author:** kdhanabalan@yahoo.com

Timeslot

The time allotment between the particular teacher and the student group for a meeting.

Class

Denotes the meeting in a particular timeslot between the student group and teacher.

Major Class

Each student belongs to a particular major like Civil Engineering, Mechanical Engineering, and Automobile Engineering etc. Each Major is considered as a Department or Discipline.

Period

Represents a particular timeslot

Meeting Conflict

Same Student group or same teacher allotted more than once at the same time.

Course

Represents a major or discipline.

Time Interval

The group of meetings which are held during the particular time period in a day.

The Set of Constraints

The set of Constraints in a timetabling problem can be stated as follows:

1. Some meetings must not be assigned to some specific timeslots.
2. Some meetings must be assigned to some specific timeslots.
3. In the timetable of each teacher, there should be idle timeslot between two meetings.
4. Required number of specific rooms are available for all meetings.
5. At any time interval more than one meeting must not be assigned to any student/teacher.
6. Saturday shall be used only after other scheduling possibilities are exhausted.
7. Some Meetings such as laboratory class may be held outside the regular timeslots if all scheduling possibilities are exhausted.
8. Any particular timeslot can be freed
9. All teachers must fulfill their weekly workload.

Here 5, 9 are physical constraints. 1, 2, 4, 8 are specification constraints. 3, 6, 7 are preference constraints.

The Problem Representation

The timetabling of an individual department of a college or for the entire college/institution can be represented as a constraint satisfaction problem (CSP). A CSP is a pair of (X,T,C) where X is a two dimensional array where row denotes the number of group of students in a department/in the entire college/institution (n) and column denotes the total number of timeslots in a week (m) for a group. T is a finite set of teachers $\{t_1, t_2, \ldots\ldots\ldots\ t_n\}$ represented by their weight. The teachers are represented by binary 2^n where n is the number of the teacher. If there are n teachers the teachers are represented as $2^1, 2^2, 2^3 \ldots\ldots 2^n$. The problem is to assign to each element of X by a value from T subject to the set of constraints C. The assignment has to satisfy all physical and specification constraints in C and minimize the number of preference constraint violations. The main objective is to minimize preference constraint violations (i.e. the number of meetings which are handled by the same teacher continuously should be minimized).If the value of a timeslot is already allotted (such as Laboratory) and if next timeslot is also allotted with the same value then that is not considered as a violation.

Proposed methods

Two different methods

1. Physical Weight Method
2. Binary Weight Method are proposed.

The steps involved in the proposed methods are explained below.

Physical Weight Method

1. Assign binary Weight to each teacher based on the number of teachers. (For i^{th} teacher it will be 2^i).
2. Create a two dimensional array X. Based on the number of student groups in a department/institution calculate the number of rows(n) and depending on the number of timeslots per day and the number of working days per week calculate the total number of timeslots in each column (m). Here each row represents a particular time interval. In a Polytechnic College there are 7 timeslots per day and 5 working days per week.
3. Fix the group and the maximum number of periods or lessons each teacher has to handle per week for each student group.
4. Generate binary weight randomly that many times equivalent to the number of periods to be handled by the teacher and fit this in a two dimensional array X.
5. Rearrange the values in the array in such a way that the physical and specification constraints mentioned earlier are met.
6. Repeat step 4 and 5 that many times equivalent to the number of populations. Let it be named as P_1, P_2, P_3, $\ldots\ldots\ldots P_r$.
7. Choose any two populations randomly (Let it be P_i, P_j where $i \neq j$ and $i, j \leq r$). Population selected in the previous selections should not be considered again.
8. Do the two point multiple years crossover with the selected populations (P_i, P_j).
   a) Choose the beginning student group randomly.
   b) Choose the ending student group randomly.
   c) Interchange the elements of the selected student groups between both the populations. After cross over process let us assume P_i, P_j are converted as O_1, O_2.

Table 1 : Science / English Department Work Allotment (Code 2)

INITIAL	MA	MB	MC	PA	PB	PC	CA	CB	CC	EA	EB
WEIGHT	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**	**2048**	**4096**
I MECH	7	0	0	4 Fixed (Computer lab)	2 Fixed (Physics lab)	3	0	2 Fixed (Chemistry lab)	3	4	0
I AUTO	0	7	0	0	2 Fixed (Physics lab)	3	4 Fixed (Computer lab)	2 Fixed (Chemistry lab)	3	4	0
I CIVIL	7	0	0	2 Fixed (Physics lab)	3	0	2 Fixed (Chemistry lab)	3	4 Fixed (Computer lab)	4	0
I CSC	0	7	0	2 Fixed (Physics lab)	3	4 Fixed (Computer lab)	2 Fixed (Chemistry lab)	3	0	0	4
I EEE	0	0	7	3	0	2 Fixed (Physics lab)	3	4 Fixed (Computer lab)	2 Fixed (Chemistry lab)	0	4
I ECE	0	0	7	3	4 Fixed (Computer lab)	2 Fixed (Physics lab)	3	0	2 Fixed (Chemistry lab)	0	4

Table 2 : Mechanical Engineering Department Work Allotment (Code 4)

INITIAL	MEA	MEB	MEC	MED	MEE	MEF	MEG
WEIGHT	**8**	**16**	**32**	**64**	**128**	**256**	**512**
II MECH	6 Lab Fixed	0	6	6 Lab Fixed	5	6 Lab Fixed	6
III MECH	0	6	6 Lab Fixed	6	6 Lab Fixed	5	6 Lab Fixed
I MECH	0	6 Engineering Drawing Fixed	0	0	0	0	0
I AUTO	0	0	6 Engineering Drawing Fixed	0	0	0	0
I CIVIL	0	0	0	6 Engineering Drawing Fixed	0	0	0
I CSC	0	0	0	0	6 Engineering Drawing Fixed	0	0
I EEE	0	0	0	0	0	6 Engineering Drawing Fixed	0
I ECE	0	0	0	0	0	0	6 Engineering Drawing Fixed

9. In each time interval there may be a possibility that same teacher weight placed more than once in O_1, O_2. This may be corrected by applying mutation.
   a) Take the i^{th} time interval and check for the existence of duplicates. If exist for each timeslot k_1 (excluding fixed slots) in the i^{th} time interval go to step b else go to step c.
   b) Check whether the same teacher is placed in the i^{th} time interval more than once.

If so interchange the particular teacher with the j^{th} time interval teacher of the same group (where $j = 1, 2, \ldots m$, $j \neq i$, m = total number of time intervals) in such a way that the interchange will not cause placing of same teacher more than once in both i and j^{th} time intervals.

Table 3: Automobile Engineering Department Work Allotment (Code 8)

INITIAL	EA	EB	EC	ED	EE	EF	EG
WEIGHT	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
II AUTO	6 Lab Fixed	0	6	6 Lab Fixed	5	6 Lab Fixed	6
III AUTO	0	6	6 Lab Fixed	6	6 Lab Fixed	5	6 Lab Fixed
I MECH	0	4 Work Shop Fixed	0	0	0	0	0
I AUTO	0	0	4 Work Shop Fixed	0	0	0	0
I CIVIL	0	0	0	4 Work Shop Fixed	0	0	0
I CSC	0	0	0	0	4 Work Shop Fixed	0	0
I EEE	0	0	0	0	0	4 Work Shop Fixed	0
I ECE	0	0	0	0	0	0	4 Work Shop Fixed

Table 4 : Civil Engineering Department Work Allotment (Code 16)

INITIAL	CA	CB	CC	CD	CE	CF
WEIGHT	**32**	**64**	**128**	**256**	**512**	**1024**
II CIVIL	5	6 + 3 Lab Fixed	6 Lab Fixed	6	6 Lab Fixed	3 Lab Fixed
III CIVIL		6 Lab Fixed	6 + 3 Lab Fixed	6 Lab Fixed	5	6 + 3 Lab Fixed

Table 5 : Computer Science Engineering Department Work Allotment (Code 32)

INITIAL	CSCA	CSCB	CSCC	CSCD	CSCE	CSCF
WEIGHT	**64**	**128**	**256**	**512**	**1024**	**2048**
II CSC	5	6 + (3 Lab Fixed)	6 Lab Fixed	6	6 Lab Fixed	3 Lab Fixed
III CSC		6 Lab Fixed	6 + (3 Lab Fixed)	0	5	6 + (3 Lab Fixed)
III EEE				6 Lab Fixed		

Table 6 : Electrical and Electronics Engineering Department Work Allotment (Code 64)

INITIAL	EEEA	EEEB	EEEC	EEED	EEEE	EEEF
WEIGHT	**128**	**256**	**512**	**1024**	**2048**	**4096**
II EEE	5	6 + (3 Lab Fixed)	0	6	6 Lab Fixed	3 Lab Fixed
III EEE		6 Lab Fixed	6 + (3 Lab Fixed)	0	5	6 + (3 Lab Fixed)
III CSC				6 Lab Fixed		
II ECE			6 Lab Fixed			

Table 7 : Electronics and Communication Engineering Department Work Allotment (Code 128)

INITIAL	ECEA	ECEB	ECEC	ECED	ECEE	ECEF
WEIGHT	**256**	**512**	**1024**	**2048**	**4096**	**8192**
II ECE	5	6 + (3 Lab Fixed)	0	6	6 Lab Fixed	3 Lab Fixed
III ECE		6 Lab Fixed	6 + (3 Lab Fixed)	6 Lab Fixed	5	6 + (3 Lab Fixed)
II EEE			6 Lab Fixed			

Table 8 : Comparison of Time and Cost

No. of generation	Time		Lowest Cost	
	Physical Weight	Binary Weight	Physical Weight	Binary Weight
50	35 Sec	36 Sec	54	49
100	70 Sec	68 Sec	46	49
200	135 Sec	122 Sec	51	55
500	344 Sec	291 Sec	51	54
1000	670 Sec	645 Sec	55	47

c) Increment i by 1 and if $i \leq m$ go to step 9(a) .

10. Repeat steps 7 to 9 till all the populations are considered.
11. If a teacher is assigned two classes consecutively then penalty cost value of 1 is added. Using this approach total penalty cost value is calculated for each population.
12. Choose the best based on the penalty value among the generated populations of $O_1, O_2, O_3 \ldots O_n$ and replace the high penalty value population with the low penalty value population.
13. Store the population and the penalty value of the best population as the p^{th} element in an array called leastcost.

The sample output for the entire institution is shown below

day1

EA	PA	PA	PA	PA	EA	MA
MEA	MEA	MEA	MEE	MEE	MEC	MEG
MEF	MEB	MEC	MEC	MEC	MEF	MED
PB	PB	EA	MB	CC	PC	CC
EG	EE	EA	EA	EA	EC	EG
EC	EC	EC	EF	ED	EB	ED
MA	MA	CA	CA	MA	MA	CB
CB	CB	CB	CB	CD	CA	CA
CC	CC	CC	CE	CB	CB	CB
MB	PC	PC	PC	PC	MB	MB
CSCB	CSCB	CSCB	CSCA	CSCD	CSCA	CSCD
CSCC	CSCC	CSCE	CSCF	CSCB	CSCB	CSCB
CB	CB	CB	CB	EB	MC	MC
EEEB	EEEB	EEEB	EEEB	EEED	EEEA	EEED
EEEF	EEEE	EEEC	EEEE	EEEB	EEEB	EEEB
EB	CA	EB	PB	PB	PB	PB
ECB	ECB	ECB	ECD	ECF	ECD	ECF
ECC	ECC	ECF	ECE	ECB	ECB	ECB

day2

PB	PB	EA	EB	EB	EB	EB
MEA	MEA	MEA	MEE	MED	MED	MED
MED	MEC	MEC	MEC	MEH	MEF	MEB
CA	CA	CA	CA	PC	EA	MB
EA	EA	EA	EC	EG	EE	EE
EG	EG	EG	EF	EC	EC	EC
PA	PA	MA	ED	ED	ED	ED
CC	CC	CC	CD	CA	CD	CD
CF	CB	CB	CB	CC	CF	CC
MB	CA	CA	MB	MB	EB	CB
CSCC	CSCC	CSCC	CSCD	CSCD	CSCB	CSCD
CSCF	CSCB	CSCB	CSCB	CSCC	CSCF	CSCC
PC	PC	EB	PA	MC	CC	CC
EEEE	EEEE	EEEE	EEEB	EEED	EEED	EEEA
EEEC	EEEC	EEEF	EEEE	EEEB	EEEB	EEEB
EB	MC	PA	EB	CA	PC	PC
ECE	ECE	ECE	ECB	ECD	ECB	ECD
ECB	ECB	ECB	ECE	ECC	ECC	ECF

day3

MA	CB	CB	MA	PC	MA	PC
MEH	MED	MED	MED	MEG	MEC	MEG
MEE	MEE	MEE	MEH	MEB	MEF	MEH
EC	EC	EC	EC	CC	CB	CB
ED	ED	ED	EG	EE	EG	EC
EB	EE	EE	EE	ED	EF	EF
CC	CC	CC	CC	EA	PB	MA
CC	CC	CC	CA	CB	CA	CB
CE	CF	CE	CD	CD	CD	CE
PA	PA	PB	EB	MEE	MEE	MEE
CSCC	CSCC	CSCC	CSCB	CSCA	CSCD	CSCB
CSCF	CSCE	CSCF	CSCF	CSCC	CSCC	CSCC
MEF	MEF	MEF	MC	CA	MC	CA
EEEE	EEEE	EEEE	EEEB	EEED	EEED	EEEB
EEEC	EEEF	EEEF	EEEC	EEEC	EEEC	EEEC
MEG	MEG	MEG	CA	PA	CC	CC
ECE	ECE	ECE	ECA	ECB	ECA	ECF
ECC	ECC	ECC	ECC	ECD	ECD	ECD

day4

MEB	MEB	MEB	CC	EA	PC	MA
MEC	MEF	MEF	MEF	MEH	MEH	MEH
MEE	MEE	MEE	MED	MEG	MEG	MEG
PC	MB	EA	MB	MEC	MEC	MEC
ED	ED	ED	EG	EF	EF	EF
EF	EE	EE	EE	EB	ED	EB
MED	MED	MED	CB	MA	CB	EA
CE	CE	CE	CB	CF	CF	CF
CC	CF	CD	CD	CD	CE	CC
EB	CB	PB	EB	MEE	MEE	MEE
CSCE	CSCE	CSCE	CSCA	CSCB	CSCB	CSCA
CSCF	CSCF	CSCF	CSCE	EEED	EEED	EEED
PA	MC	EB	CA	MEF	MEF	MEF
EEEF	EEEF	EEEF	EEEA	ECC	ECC	ECC
EEEE	EEEC	EEEE	CSCD	CSCD	CSCD	EEEF
MEG	MEG	MEG	PA	MC	MC	MC
ECF	ECF	ECF	ECB	EEEC	EEEC	EEEC
ECE	ECD	ECD	ECD	ECE	ECF	ECF

day5

MEB	MEB	MEB	MA	CC	CC	MA
MEF	MEF	MEF	MEE	MEH	MEH	MEE
MEH	MEH	MEH	MEG	MEG	MEG	MEF
MB	EA	MB	MB	MEC	MEC	MEC
EE	EC	EC	EC	EF	EF	EF
EG	EG	EG	EB	ED	ED	EB
MED	MED	MED	EA	PB	EA	PB
CD	CB	CE	CE	CE	CD	CB
CC	CF	CF	CF	CC	CF	CF
PB	EE	EE	EE	EE	CB	MB
CSCF	CSCF	CSCF	CSCB	CSCE	CSCE	CSCE
CSCE	CSCE	CSCC	EEED	EEED	EEED	CSCC
EF	EF	EF	EF	MC	EB	PA
EEEB	EEEA	EEEA	EEEB	ECC	ECC	ECC
CSCD	CSCD	CSCD	EEEF	EEEF	EEEF	EEEF
MC	MC	MC	EG	EG	EG	EG
ECB	EEEC	EEEC	EEEC	ECD	ECD	ECA
ECF	ECF	ECF	ECC	ECE	ECF	ECF

If interchange causes placing of same teacher more than once in any one of the i and j^{th} time interval then increment j by 1 and if $j < m$ go to step c else the solution is infeasible and go to step 8.

14. Increment the number of generations by 1 and increment p by 1.
15. Repeat steps from 7 to 11 until the required number of generations are generated.

16. Choose the best population among the stored populations in leastcost array.

Binary Weight Method

While doing steps 9a and 9b in physical weight method there will be more number of physical comparisons has to be carried out .

1. to find whether a particular element is present in the given set of elements
2. and to find the existence of duplicate element in the given set of elements.

In Binary weight method the following two different approaches are handled while doing steps 9a and 9b. Suppose if there are set of elements S_1, S_2, S_3 ….. S_n and to search for any particular element say S_x in the above set a maximum of n comparisons are to be made. If all the elements are binary weighted, bitwise arithmetic OR operation can be performed on elements of the set and a value Q_1 can be obtained. To this Q_1, bitwise arithmetic OR operation can be done with S_x and store it as Q_2. If both Q_1 and Q_2 are equal then search element is present otherwise search element is not present. To find if any of the element is duplicated in a set of elements S_1, S_2, S_3 ….. S_n, if all the elements are binary weighted then

a) Find the sum of the decimal values of the ($S_1 + S_2 + S_3 +$ …….. S_n) and store in Q_3
b) Perform arithmetic bitwise OR operator on (S_1, S_2, S_3 ….. S_n) and store the decimal value of the result in Q_4

If Q_3 and Q_4 are not equal then some of the elements are duplicated otherwise no element is duplicated.

EXPERIMENTAL RESULTS

Tests were done on the real life data of a Polytechnic College. Our main intention is assignment of classes for the entire college. Since we are giving binary weight to teachers there is a limitation in generation of binary weight based on the system configuration. Initially one department is processed as given in the example, then while processing next department already allotted meetings of previous department teachers considered as fixed and one similar weight is given to all processed teachers belong to the same department so that the already assigned weights can be reused for next group. In this manner all departments are processed sequentially one by one. The work allotment and the weight allotment of each department and the respective student group is tabulated below. If there are any Lab hours those are treated as Fixed and allotment carried out accordingly. HCL machine having P IV Intel processor with 1GB main memory and Microsoft visual studio version 2005 Vb.net is used to run all tests. It is found that if number of duplicates are less then binary weight method is faster than physical weight method. Constraint violations need to be reduced further.

Conclusions

In this we have proposed two different methods one using binary weight and another one using ordinary weight. A new kind of sequential evaluation process including selective two point multiple years crossover and mutation is introduced. Both the methods proved to useful in course timetabling problem of entire institution or group of departments. Currently working on another mutation process which is reducing the penalty cost and handling multiple objective functions and also working on Engineering College Timetabling and also working on exam timetabling using this similar approach. Also working on processing all groups in parallel and running all at the same time without grouping.

REFERENCES

1. Geraldo Ribeiro Filho and Luiz Antonio Nogueira Lorena A Constructive Evolutionary Approach to School Timetabling, in Application of Evolutionary Computing, Lecture Notes in Computer Science, vol. 2037, Springer – Verlag, 2001, page 130 – 139.
2. M. Akif Bakir and Cihan Aksop, A 0 – 1 Integer Programming Approach to a University Timetabling Problem, Hacettepe Journal of Mathematics and Statistics Volume 37 (1) (2008), 41 – 55.
3. Maciej Norberciak, Universal Method for Timetable Construction based on Evolutionary Approach, International Journal of Applied Mathematics and Computer Sciences Volume 3 Number 3, 2006.
4. Burke E.K., Elliman D.G. and Weare R.F. (1994) "A Genetic Algorithm Based University Timetabling System", AISB Workshop on Evolutionary Computing.
5. Supachate Innet, Nawat Nuntasen, University Timetabling Using Evolutionary Computation, Wseas Transactions on Advances in Engineering Education Manuscript received Oct 18, 2007, revised Dec. 28, 2007.
6. Malek Rahoual, Rachid Saad, Solving Timetabling Problems by Hybridizing Genetic Algorithms and Tabu Search,PATAT 2006, pp. 467–472. ISBN 80-210-3726-1).
7. Cagdas Hakan Aladag and Gulsum Hocaoglu, A Tabu search Algorithm to solve a course Timetabling Problem, Hacettepe Journal of Mathematics and Statistics Volume 36 (1) (2007), 53 – 64
8. Haroldo G. Santos , Luiz S. Ochi , Marcone J.F. Souza, A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem, Journal of Experimental Algorithmics (JEA), 10, 2005.
9. Pawel Myszkowski, Maciej Norberciak, Evolutionary algorithms for timetable problems, Annales UMCS Information AI 1 (2003) 115 – 125, http://www.annales.umcs.lublin.pl/.
10. Christelle Guéret, Christian Prins, Marc Sevaux: Applications of optimization with Xpress-MP c 2000 Editions Eyrolles, Paris, France.Translated and revised by Susanne Heipcke.]
11. S. Gyori, Z. Petres, A. R. Varkonyi – Koczy, Genetic Algorithms in Timetabling. A New Approach, 2001, http://www.mft.hu/hallg/200107.pdf.
12. Duong Tuan Anh, Vo Hoang Tam, and Nguyen Quoc Viet Hung, Combines Interactive and Automatic Course Timetabling, Citeseer - 2008.
13. A Framework for School Timetabling Problem Kimmo Nurmi, Jari Kyngäs, MISTA 2007.
14. Aldy Gunawan, K.M. Ng, H.L.Ong, A Genetic Algorithm for the Teacher Assignment Problem for a University in Indonesia, Information and Management Sciences, Volume 19, Number 1, pp. 1 – 16, 2008.

15. Rhydian Lewis and Ben Paechter, An Emprical Analysis of the Grouping Genetic Algorithm: the Timetabling Case, The 2005 IEEE Congress.
16. A Parallel Genetic Algorithm for solving the School timetabling problem by D. Abramson J. Abela, High Performance Computation Project Appeared in 15 Australian Computer Science Conference, Hobart, Feb 1992, pp 1-11.
17. A genetic algorithm solving a weekly course-timetabling problem Wilhelm Erben and Jürgen Keppler,1995.
18. An Application of Genetic Algorithms to Course Scheduling at the United States Army War College by M A J James J. Donlon, pringer Berlin/Heidelberg,2001

****