Performance Analysis of Diversity Measure with Crossover Operators in Genetic Algorithm

M.Nandhini
Research Scholar
Research and Development Centre
Barathiar University
Coimbatore, India-46

S.Kanmani
Prof. & Head
Department of Information Technology
Pondicherry Engineering College
Puducherry, India-14

S.Anandan
Department of Computer Science
Pondicherry University
Puducherry, India-14

ABSTRACT
The goal of np-hard Combinatorial Optimization is finding the best possible solution from the set of feasible solutions. In this paper, we establish an approach using genetic algorithm with various selection and crossover operators with repair function for an institute course timetabling problem. It employs a constructive heuristic approach to find the feasible timetable, fitness value calculation, selection operators, crossover operators and repair function. The performance of proposed and existing selection and crossover operators are compared and shown by keeping diversity in the fitness value of population.

General Terms
Heuristics, Evolutionary Computation, Genetic Algorithm, Artificial Intelligence.

Keywords
Course timetabling, fitness, selection, crossover, repair, optimal solution.

1. INTRODUCTION
Timetabling is one of the common problems of scheduling which can be described as the allocation of resources under predefined constraints so that it maximizes the possibility of allocation or minimizes the violation of constraints. The constraints are classified as hard or soft. Hard constraints are those to which a timetable has to adhere in order to be feasible. A timetable is not viable if it does not satisfy them. Soft constraints are also those whose violation should be minimized. Practical timetabling problems have many forms like educational timetabling (course and exam), employee timetabling, timetabling of sports events, timetabling of transportation means, etc.

Timetabling as well as scheduling problems, define a class of hard-to-solve constrained optimization problems of combinatorial nature. Such problems are mainly classified as constraint satisfaction problems [3], where the main goal is to satisfy all problem constraints, rather optimizing a number of objectives.

Still in some institutions, trial and hit method is used due to its cumbersome feature. The person in charge for the scheduling will take the challenge of preparing a new timetable with reference to previous timetables. Using computational algorithms, to automate timetabling is of great importance as it can save a lot of man-hours work, to institutions and companies, and provide optimal solutions with constraint satisfaction that can boost productivity.

On computing nature, there are various Artificial Intelligence approaches like Hill Climbing(HC), Simulated Annealing(SA), Tabu Search(TS) and Genetic Algorithm (GA) are used [7] and applied these techniques to solve specific variants of the timetabling problem such as school timetabling, course timetabling, and examination timetabling problem. The problem faced by SA is that it can't escape from local minima once the temperature has become very low. Similarly in TS, the algorithm takes long time when the Tabu list increases.

Among the various traditional techniques, GA is focused specially because of its parallel nature of stochastic search, less likely to get struck at local minima and less sensitive to initial conditions[6].

The College Course Timetabling (CCTP) is one of the common educational timetabling problems which can be seen as a part of the University course timetabling . Since CCTP is NP-hard, a variety of approaches have been adopted, achieving varying levels of success. CCTPs are search problems, in which subjects must be arranged around a set of time slots, so as to satisfy given constraints and optimize a set of objectives [5]. The NP-hard class problems are very difficult to solve using conventional optimization techniques. In evolutionary algorithms, a solution with better fitness could be obtained with minimum explorations.

In this paper, a constructive heuristic approach is employed to find the feasible timetable. Its fitness is evaluated from the penalty cost raised due to the violation of soft constraints. The performance of GA is tested with two types of selection and two types of crossover operators with repair function and thereby determining better operators favoring the problem. The performance evaluation is done with genetic parameters namely, generation, population size, crossover rate, fitness value. This methodology is implemented for a weekly scheduling of lectures of Bachelor of Technology course in the Department of Information Technology of Pondicherry Engineering College as an illustration.

This paper contains four major sections. Section.2 and Section.3 describe survey on this work and the implementation of automated course timetabling using GA with crossover respectively. Section.4 reviews on various results obtained and finally, Section.5 concludes the paper.

2. LITERATURE SURVEY
Abdelaziz Dammak, Abdelkarim Elloumi and Hichem Kamoun(2009)[1] developed lecture timetabling at a Tunisian university and thereby applied heuristics procedure to construct a

feasible timetable for all the lectures taken by the students sections in the institution.

Salwani Abdullah, Hamza Turabieh(2008)[11] generated University course timetable using genetic algorithm and local search, in which single point crossover has been applied. This operation performs single point recombination between pairs of chromosomes and returns the parent chromosomes after mating. In this ,crossover point is chosen randomly from the domain (1,2, 3, …, L) where L is chromosome length. In the chromosomes, genes before the crossover point are kept unchanged and the genes after the crossover position are swapped. This work resulted with better performance than earlier works.

Wutthipong Chinnasri, Nidapan Sureerattanan(2010)[14] performed the comparison between different selection strategies on genetic algorithm with course timetabling problem and proved that roulette wheel selection works better than rank and tournament selections.

Salwani Abdullah, Edmund K.Burke and Barry McCollum (2007)[10] described a hybrid evolutionary approach to the University Course Timetabling Problem . In this mutation was done by selecting a course at random and making changes without violating feasibility

Ali K. Kamrani and Ricardo Gonzalez(2008)[2] developed genetic algorithm based solution approach to combinatorial optimization problems with depth first branch and bound algorithm and the local search with GA and received better results.

Rhydian Lewis and Ben Paechter. (2007)[9] designed a method for measuring population diversities and distances between individuals with the grouping representation. It improved the performance of GA by the introduction of a number of different fitness functions and the use of an additional stochastic local-search operator.

Panagiotis Adamidis and Panagiotis Arapakis (1999)[8] used elitism, and both recombination and mutation with adaptive operator probability. It made adjustment in operator probabilities, depending on the convergence of the population.

Kremena Royachka and Milena Karova (2006)[6] used random walk selection and adaptive threshold mutation operators and resulted with better result. Ghaemi, S., Vakili. M.T. and Aghagolzadeh.A Ghaemi (2007)[5] employed Modified GA and Cooperative GA and shown better results with modified basic genetic operators and used intelligent operators and cooperative genetic method to improve overall algorithm's behavior. Wang Xiao Yun, Wang Feng Kun and Wang Xiang Yun (2008) [13] developed Improved GA and found effective exploitation of search space with dynamic variation select rate and elitist strategy with dissimilarity chromosome.

Yu Zheng, Jing-fa Liu, Wue-hua Geng and Jing-yu Yang (2009)[15] proposed a novel quantum-inspired evolutionary algorithms (QEA) which is put forward for the CTP and proved its significance in convergence rate and in providing high quality tables.

Edmund Burke, Jakub Mareček, Andrew J.Parkes and Hana Rudová(2010)[4] applied decomposition, reformulation, and diving in university course timetabling with the approach of multiphase exploitation of multiple objective-/value-restricted submodels.

The performance of diversity is measured with different combination of selection and crossover operators. Mating of two different distinct (best and worst) featured individuals may result with good offspring and this led us to propose selection operator called grade. In PMX, due to context sensitivity of course timetables, crossover against two individuals matching section, mostly forms individuals by doing minimum swapping. This reduces the diversity in resulting population. This led us to propose a crossover operator called combinatorial PMX.

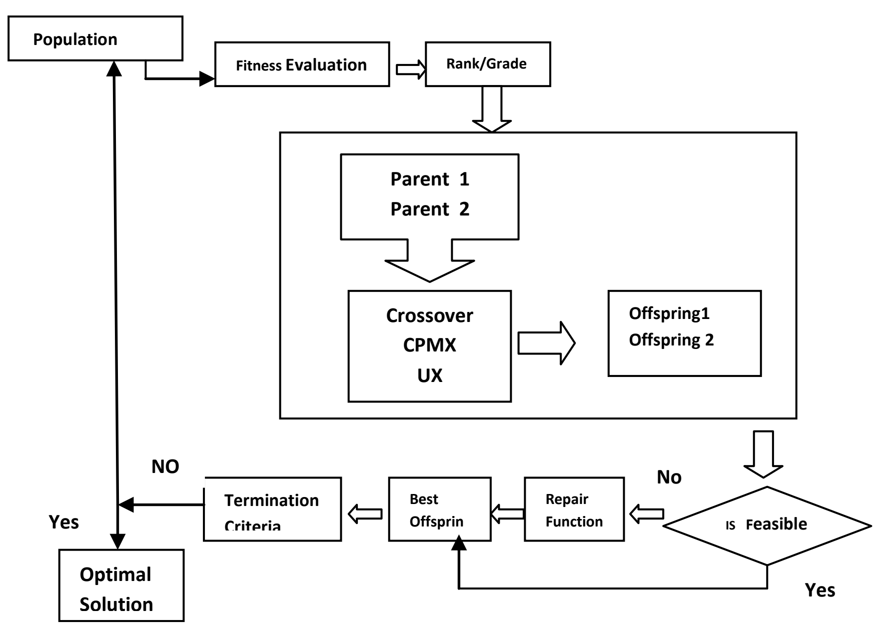

Fig. 1: Block diagram of proposed GA architecture with crossover and selection operators

3. IMPLEMENTATION OF AUTOMATED COURSE TIMETABLING USING GA WITH CROSSOVER

In this paper, the performance of GA is experimented with two types of selection viz., Rank and Grade and two types of crossover operators viz., Uniform and Combinatorial partially matched . Problem description is given in the annexure A and the proposed GA architecture is shown in Figure 1.

3.1 Initial Population

The initial population consists of a number of chromosomes equal to the population size. Each chromosome is created using the constructive heuristic approach and is represented as a three-dimensional matrix. Lower index of the matrix represents period, middle index represents day and upper represents a class. The value of each cell (timeslot) of the matrix represents allotment scheduled in the corresponding class and period. The initialization procedure in Figure 2 encodes the input data into chromosome representation. The initialization process obtains feasible chromosomes that satisfy several hard constraints which may be as follows.

```
for each chromosome
    begin
        for each class
        begin
            for each Practical subject
            begin
                Make entry for continuous time slots in either
                of the sessions other than first period and
                without room conflicts
            end
            for each theory subject
            begin
                Make entry for all periods in class and
                teachers timetable without violating hard
                constraints
            end
        end
end.
```

Fig.2: Population Initialization Procedure

3.1.1 Hard constraints

Subject Conflict

- More than one period in a day cannot be assigned for one subject.

Student Conflicts

- No student can be assigned more than a course at the same period.

Teacher Conflicts

- No teacher can be scheduled for either two classes or one class and a lab at the same period.
- Maximum workload of a teacher must not be exceeded.

RoomConflicts

- Laboratory periods for different classes assigned in a physical laboratory location must not overlap.
- Laboratory periods should come in the continuous times lot either in the morning or in the evening session but not in the first period of both sessions.

3.1.2 Soft constraints

- At least one period gap would be given between the lecture periods of a teacher in a day.
- In adjacent days, two same periods would not have the same subject.
- First period of a day would be different from other day.
- Each staff would be given first period at least once in a week.
- Free periods would come in the afternoon session.
- Each teacher can be assigned maximum of 2 theories/ one theory and one lab/ 2 theories and one practical only in a day.

3.1.3 Fitness function

A factor to evaluate the timetable for finding its level of optimality is fitness function. This is evaluated from the penalty cost and validity of soft constraints. A chromosome having minimum fitness value in the population is the best solution.

$$Min\ f(T) = \sum_{j=1}^{n\in SC} p(j)V(j)$$

Where:

$p(j)$ = Penalty cost of soft constraint j on T.

$v(j)$ = Validity of Soft constraint j .

T = Timetable

SC = Soft Constraint

If $j \in SC$ on T is satisfied, then $v(j) = 0$,

Otherwise $v(j) = 1$.

3.1.4 Elitism Strategy

Elitism is a method, which copies few best chromosomes into new population. The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents loosing the best found solution.

3.2 Selection

Selection is the process of choosing parents from the generated population to undergo genetic operations like mutation or crossover. Two selection operators viz.,rank and proposed one of grade are applied.

3.2.1 Rank selection

Rank selection is used to form a mating pool of M solutions from the population. Chromosome having minimum fitness is assigned with higher rank. The higher rank(worst) solutions are taken for improving them in the successive steps.

Procedure for Rank Selection

Sort the initial population based on fitness value.

Select two chromosomes i1,i2 having low fitness value(higher rank).

Have i1 as Parent1 , i2 as Parent2.

3.2.2 Grade selection

Diversity in population helps to get global optimum. In order to increase diversity, this selection operator is proposed which takes chromosomes from variety of groups randomly and mating those chromosomes result in salient features.

Procedure for Grade Selection

Find the standard deviation for the individuals in the mating pool using the formula,

$$SD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

Where

x_i - Fitness of the i^th individual

N - Number of individuals in the mating pool

$\overline{x}$ - Average cost of fitness.

SD decides the range of values in a group for a grade. Divide the mating pool into groups with grades by fixing costs in the range ($\overline{x}$ - SD , $\overline{x}$ + SD) as average grade, and form range of higher grades by adding SD with average and form lower grades by deducting SD from the average. Steps to be performed to select offspring are,

- Select the first parent randomly from any one group
- Select the second parent randomly from any group other than the one containing first parent
- After selecting both the parents, remove them from the mating pool.

As a result, parents would be selected from any two different grade (nature) of groups that could have created offspring with diverse in nature. Parents are selected from different groups such as worst and better, worst and good, better and better combinations and having higher possibility of producing better offspring and thus diversity could be improved.

3.3 Crossover

Crossover operator aims to interchange the information and genes between chromosomes. Therefore, crossover operator combines two or more parents to reproduce new children, then, one of these children may hopefully collect all good features that exist in his parents. On combining inversion and crossover, the reordering operators proposed are[12],

Combinatorial Partially Matched Crossover (PMX) and Uniform Crossover (UX).

3.3.1 Crossover rate

Crossover occurs during evolution according to a user-definable crossover probability and fixed as 0.8.

3.3.2 Partially matched crossover(PMX)

Partially matched crossover (PMX) may be viewed as a crossover of permutations, which guarantees that all positions are found exactly once in each offspring, i.e. both offspring receive a full complement of genes followed by the corresponding filling in of alleles from their parents.

PMX proceeds as follows:

1) The two chromosomes are aligned.
2) Two crossing sites are selected uniformly at random along the strings, defining a matching section.
3) The matching section is used to effect a cross through position by position exchange operation.
4) Alleles are moved to their new positions in the offspring.

3.3.3 Combinatorial partially matched Crossover (Combinatorial PMX)

Due to the context sensitivity of this problem, altering timeslots within chromosomes result with uncertainty. This led us to propose a crossover operator which is a variant of PMX named as Combinatorial Partially Matched crossover. Wherein possible combinations of matching section are formed by altering position of timeslots. Crossover is done for all combinations of matching sections individually. One of the combinations of matching section produces offspring by doing maximum gene exchanges is considered for further operations. Mating is done between respective classes of two chromosomes. In turn, in each class, combinatorial PMX applied on days with placement and training (tnp) and seminar/group discussion(seminar/gd) and the procedure is shown in Figure 3.

```
Repeat
  Repeat
  Select two parents P1,P2. [ Based on rank and grade
  selection];
  Select partial set of slots in the days containing GP and T&P
  hours;
  Generate combinations for the selected slots;
    Repeat
      Repeat
       Case :1 : Find the teacher T1 for the slot X1 in P1;
                 Find the teacher T2 for the slot Y1 in P2;
                 if (T2(position of slot X1) and T1(position of slot
                Y1) is free)
                Swap;
                Count ++;
                end if;
      Case :2 : Find the teacher T1 for the slot  X1 in P1,
                          Slot Y1 in P2 is free;
                 if(T1(position of slot Y1) is free
                 Swap;
                 Count ++;
                 end if;
      Until( all  slots in partial sets are checked);
    Until ( all combinations are attempted);
Select the combination having maximum(count);
Normalize each subject workload in the child;
         if (child is feasible)
         Replace parent by child;
         else if(child is infeasible)
         Apply repair function ;
         end if;
Until(crossover rate is reached);
Until(termination criteria is met );
```

Fig. 3: Combinatorial PMX

3.3.4 Uniform crossover

Uniform crossover[12] is quite different from the N-point crossover. Each timeslot in the offspring is created by copying the corresponding timeslot from one or the other parent chosen according to a randomly generated binary crossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the timeslot is swapped from the first parent to the timeslot in the corresponding second parent. If there is a 0 in the crossover mask , no swapping takes place. A new crossover mask is randomly generated for each pair of parents. Offspring, therefore contain a mixture of genes from each parent
and the procedure is explained in Figure 4.

```
Repeat
  Repeat
      Select two parents P1,P2. [Based on rank and grade
      Selection];
      Select all the slots except GP and T&P slots in the days
                                   containing GP and T&P hours;
    Generate random mask bits{0,1} for the selected slots from
                                   the parents P1 and P2;
      Repeat
      if (mask bit equals 1)
            Find the teacherT1 for the slot X1 in P1;
            Find the teacherT2 for the slot Y1 in P2;
            if (T2(position of slot X1) and T1(position of slot X2)
            is free)
             Swap the slots;
            end if
       else if(mask bit equals 0)
                    check for next slot;
       end if
      Until(all mask bits are processed);
   Until(crossover rate is achieved);
Until(termination criteria is met);
```

Fig. 4: Uniform Crossover

3.4 Repair function

Repairing is mainly done on violation of hard constraints. Knowing the location of the offending timeslots, replace these timeslots iteratively with valid timeslots. Violation of teacher conflict and subject conflict are rectified in the offspring attained after mating. Procedure of different possible situations raised during repairing is given in Figure 5.

```
Repair Procedure for Teacher's conflict
/* Case.1. Swapping Free timeslot with Theory Timeslot of
same day*/
Repeat
  Find timeslots TS1<- free; TS2 <- Theory ; TS3<- Theory in a
                                              day Di of Class Ci
   To swap(TS1,TS2);
        If Teacher(TS2) not free in TS1
        {
               Swap(TS1,TS3);
               Swap(TS2,TS1);
        }
        Else if
               Swap(TS1,TS2);
        End if;
Until(Repairing all Teacher's Conflict);
/*Case.2. Swapping Free timeslot with Theory Timeslot of
same day with intern adjustment*/
Repeat
Find TS1<-free; TS2<- Theory in a day Di of Class Ci
To swap (TS1, TS2);
   If Teacher(TS2) not free in TS1
   {
     Make free Teacher(TS2) in TS1 by adjusting in the class
     Ci+1/ Ci+2;
     Swap(TS1, TS2);
    }
Until(Repairing all Teacher's Conflict);

/*Case. 3 . Swapping Free timeslot with Theory Time slot of
some other day*/
Repeat
Find TS1<-free ; TS2<- Theory in a day Di of Class Ci
To swap (TS1, TS2);
If  Teacher(TS2) free in TS1
    {
          If (!subject's conflict)
                 Swap(TS1,TS2);
          Else
          {
          Normalize subject's conflict;
          Swap(TS1,TS2);
          }
   }
Until(Repairing all Teacher's Conflict);

/*Repair procedure for Subject's conflict*/
/*A subject comes more than once in a day*/
Repeat
 Find the subject S1 occurring more than once in a day  D1;
 Case 1: Try o replace to free timeslot in a day where S1 not
         occurs;
         If  Teacher(S1) not free in the free timeslot
         Repair Teacher's conflict and replace;
  Case 2 : Find a day where S1 not occurs;
          Swap with any timeslot of  S2, provided S2 not in D1;
Until(Repairing all subjects conflict)
```

Fig.5: Repair Procedure

3.5 Termination Criteria

This iterative process continues until one of the possible termination criteria is met. The possible termination criteria are reaching optimal, getting acceptable solution level, performing maximum number of generations and moving on generations without any improvement in fitness value.

4. EXPERIMENTAL RESULTS

This algorithm has been implemented using Java. This algorithm is tested with a standard timetable requirements specified in annexure A. It has been observed that the timetable created by this algorithm is more optimal than one which is created manually.

Automated CCTP has been tested with population sizes 100, 200,400 and 600 for generation up to 600, with different datasets. From the experimental results of different combination of selection and crossover operators viz., rank with UX, rank with CPMX, grade with UX and grade with CPMX with 80% of crossover probability rate , their performance on same population in different generation is given in Table 1.

Performance of diversity measure is evaluated from the range of fitness. The inspiration is due to diversity, range of fitness increases as generation grows.

In Figure 6, pair of lines denotes the range of fitness for different population sizes of each pair of operators. From the results of implementation, it is identified from Table 1 and

Table 1. Performance Study Of Selection And Cross Over Operators

Population	Generation	Rank + UX		Rank + Combinatorial PMX		Grade + UX		Grade + Combinatorial PMX	
100	1	1350	4950	1350	4950	1350	4950	1350	4950
	5	1350	4950	1350	4950	1350	4950	1350	4950
	20	1100	4950	1350	4950	1350	4950	1350	4950
	50	1100	4950	1350	4950	1100	4950	1350	4950
	75	1100	4750	1350	4950	1100	4950	1150	4950
	100	1100	4600	1350	4800	1100	4850	1050	4950
200	1	1200	5000	1200	5000	1200	5000	1200	5000
	5	1200	5000	1200	5000	1200	5000	1200	5000
	20	1200	5000	1200	5000	1200	5000	900	5000
	50	1100	5000	1200	5000	1200	5000	900	5000
	100	1100	5000	1200	5000	1200	5000	900	5000
	150	1100	4950	1200	4800	1200	5000	900	5000
	200	1100	4200	1200	4800	1200	4900	900	5000
400	1	1050	4900	1050	4900	1050	4900	1050	4900
	20	1050	4900	1050	4900	900	4900	1050	4900
	50	1000	4900	1050	4900	750	4900	1050	4900
	100	1000	4900	1050	4900	750	4900	1050	4900
	200	1000	4900	1050	4900	700	4900	1050	4900
	250	1000	4900	1050	4900	700	4900	1050	4900
	300	1000	4900	1050	4900	700	4900	1050	4900
	350	1000	4800	1050	4800	700	4900	1050	4900
	400	1000	4700	1050	4800	700	4900	1050	4900
600	1	1100	4600	1100	4600	1100	4600	1100	4600
	50	1100	4600	1100	4600	700	4600	750	4600
	100	1100	4600	1100	4600	700	4600	750	4600
	200	1100	4600	1100	4600	700	4600	750	4600
	250	1100	4600	1100	4600	700	4600	750	4600
	300	1100	4600	1100	4600	700	4600	750	4600
	350	1100	4400	1100	4600	700	4600	750	4600
	400	1100	4200	1100	4400	700	4600	750	4600
	500	1100	4200	1100	4350	700	4600	750	4600
	600	1100	4200	1100	4300	700	4600	750	4600

Figure 6 that grade selection is performing better than rank selection by retaining diverse search space. While parsing generations, the search space fitness reduction exists in rank selection.

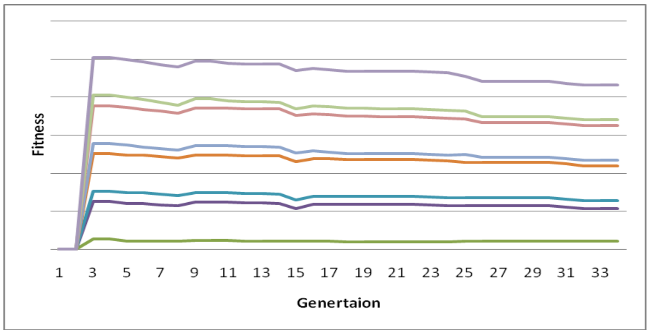

Fig 6: Performance Comparison of all Selection and Crossover Operators

In crossover operators, our proposed combinatorial partially matched crossover is performing better by keeping the fitness range apart in both rank and grade selection and is shown in Table 2.

Table 2: CPMX with selection operators

Population	Generation	Fitness Range Difference: Rank with CPMX	Fitness Range Difference: Grade with CPMX
100	100	3450	3900
200	200	3600	4100
400	400	3750	3850
600	600	3200	3850

Grade with CPMX decides the optimality factor by keeping diverse search space and by producing near optimal values. With this , it would be concluded that grade with CPMX could produce more promising results.

5. CONCLUSION

With this, significance of crossover while finding global optimal in maintaining diversity has been identified and found the better combination of selection and crossover operators in keeping the fitness value range with less reduction. It is concluded that combination of grade with combinatorial PMX is working better. As further enhancement, this might be combined with mutation and local search to get the better optimal with efficient combination of genetic operators.

6. REFERENCES

[1] Abdelaziz Dammak, Abdelkarim Elloumi and Hichem Kamoun. 2009. Lecture timetabling at a Tunisian university, InderScience. Int. J. of Operational Research.4(3) (2009), 323 - 345.

[2] Ali K. Kamrani and Ricardo Gonzalez. 2008. Genetic algorithms-based solution approach to combinatorial optimisation problems. Inderscience, Int. J. Knowledge Management Studies, 2(4) (2008) , 499 - 518.

[3] Brailsford, C., Potts, C. N., and Smith, B. M. 1999. Constraint satisfaction problems: Algorithms and applications. European J. Operational Research. 119(1999),557–581.

[4] Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudova. 2010. Decomposition, reformulation, and diving in university course timetabling . Computers & Operations Research, 37(3) (2010), 582-597.

[5] Ghaemi, S., Vakili, M.T., Aghagolzadeh. A. 2007. Using a genetic algorithm optimizer tool to solve university timetable scheduling problem. In Proceedings of IEEE 9th Int. Symposium on Signal Processing and its Applications(2007),1 – 4.

[6] Kremena Royachka and Milena Karova. 2006. High-Performance Optimization of Genetic Algorithms. In Proceedings of IEEE Int. Spring Seminar on Electronics Technology(2006) ,395-400.

[7] Nguyen Due Thanh. 2007. Solving Timetabling Problem Using Genetic and Heuristic Algorithms. In Proceedings of 8th ACIS Int. Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 3(2007),472 – 477.

[8] Panagiotis Adamidis and Panagiotis Arapakis. 1999. Evolutionary Algorithms in Lecture Timetabling . In Proceedings of Congress on Evolutionary Computation , 2(1999),1145-1151.

[9] Rhydian Lewis , Ben Paechter. 2007. Finding feasible timetables using group-based operators. IEEE Transactions on Evolutionary Computation, 11(3)(2007),397-413.

[10] Salwani Abdullah, Edmund K. Burke, and Barry McCollum. 2007. A hybrid evolutionary approach to the University Course Timetabling Problem. In Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007), 1764-1768.

[11] Salwani Abdullah and Hamza Turabieh. 2008. Generating University Course Timetable Using Genetic Algorithms and Local Search. In Proceedings of IEEE 3rd Int. Conference on Convergence and Hybrid Information Technology, (2008), 254-260.

[12] Sivanandham, S. N., and Deepa, S. N. 2008. Introduction to Genetic Algorithm , Springer .

[13] Wang Xiao Yun , Wang Feng Kun and Wang Xiang Yun. 2008. Optimize Timetabling Problem Using Improved Genetic Algorithm. In Proceedings of IEEE Workshop on Knowledge Acquiring Modeling, (2008), 260-262.

[14] Wutthipong Chinnasri, Nidapan Sureerattanan .2010. Comparison of Performance Between Different Selection Strategies on Genetic Algorithm with Course Timetabling Problem. In Proceedings of Int. Conference on Advanced Management Science, (2010),105-108.

[15] Yu Zheng, Jing-fa Liu, Wue-hua Geng and Jing-yu Yang, 2009. Quantum-Inspired Genetic Evolutionary Algorithm for Course Timetabling. In Proceedings of 3rd Int. Conference on Genetic and Evolutionary Computing, 750-753.

ANNEXURE. A

Problem Description

The description of timetabling of the Bachelor of Technology Course offered in the Department of Information Technology, Pondicherry Engineering College is as follows.

The Course contains 4 classes (each for a year of study). The framework of each B.Tech course in the Institute is of the form 5 (days) * 8 (periods). Timeslots represents intersection of day and period. In each day, morning and afternoon session has four periods. Each course has six theory subjects and three laboratory subjects. Each theory subject should be allotted with four timeslots and practical subject with 3 continuous periods in a week. Due to room conflict, each practical will be conducted for 3 days by dividing students into 3 batches. Thereby, each practical should be monitored by a staff for nine periods. Co-curricular activities such as placement and training for 3 periods, seminar / group discussion for 2 periods must be allotted for each class. The parameters required to design the timetable is shown in Table 3.

Table 3. Parameters Specification

No.	Description	Quantity
1	No. of classes	4
2	No. of Maximum Theory Subjects per Class	6
3	No. of practical per Class	3
4	No. of timeslots/ theory	4
5	No. of timeslots / practical	3
6	No. of Teachers	12
7	No. of days	5
8	No. of timeslots in a day	8
9	No. of placement and training periods	3
10	No. of seminar/ group discussion periods	2
11	No. of free periods	2
12	Total hours per week (including free periods)	40