Available online at www.elixirpublishers.com (Elixir International Journal)

Advanced Engineering Informatics

Elixir Adv. Engg. Info. 37 (2011) 3678-3690

ISSN: 2229-712X

Hybrid approach for solving a combinatorial problem with gene tuning

M. Nandhini[1], S.Kanmani[2] and S.Anandan[3]

[1]Research and Development Centre, Bharathiar University, Coimbatore-46, TamilNadu, India.
[2]Department of Information Technology, Pondicherry Engineering College, Puducherry-14, India.
[3]Department of Computer science, School of Engineering and Technology, Pondicherry University, Puducherry-14, India.

ARTICLE INFO

Article history:
Received: 1 June 2011;
Received in revised form: 16 July 2011;
Accepted: 26 July 2011;

Keywords
Course timetabling,
Grade selection,
Rank selection,
Combinatorial partially matched crossover,
Mutation,
Gene tuning,
Local search.

ABSTRACT

A new hybrid approach of Genetic Algorithm (GA) with local search algorithm is proposed to solve Course Timetabling Problems (CTP). In which, GA architecture is enhanced by proposing various selection, crossover and mutation operators. Diversity in population helps to get global optimum. In order to accommodate diversity of population and to avoid local optima, grade selection and combinatorial partially matched crossover operators are proposed. To increase the convergence rate and to produce guaranteed result, various mutation strategies are proposed with gene tuning approach. To improve the quality of the solution, steepest ascent hill climbing local search algorithm has been proposed. With these, hybrid approach with enhanced GA is implemented on CTP and hence its quality is proved by getting more promising and consistent results in all operations of the possible twelve combination of GA proposed operators. Also, proved experimentally that combination of grade selection, combinatorial partially matched crossover and adaptive mutation strategy operators is performing the best among all twelve proposed combinations and a combination of operators from the literature by yielding the average relative convergence rate as 31% which is greater than all others' convergence rate.

Introduction

Course Timetabling Problem (CTP) is an NP hard combinatorial problem, which is very difficult to solve by conventional methods. The amount of computation required to find optimal solution increases exponentially with the problem size. The main objective of this problem is scheduling a sequence of lectures between teachers and students in a pre-fixed period of time typically a week, satisfying a variety of constraints/objectives. In general, the constraints are of two types - hard constraints, which are conditions that must be satisfied strictly; soft constraints, which are the ones that may be satisfied as much as possible. It is therefore necessary to use efficient search methods to produce optimal or optimal near timetable that satisfy the constraints. A large number of diverse methods have been proposed so far in the literature for solving timetabling problems. These methods come from a number of scientific disciplines such as Operations Research, Artificial Intelligence, Computational Intelligence can be broadly classified into four categories.Abramson.D(1991),Burke.E.K. et.al(2002),Hertz.T.A(1991),Paechter.T.B.et al.(1995), Schaerf.T.A (1993) & Tripathy. T.A(1980).

• Timetabling problems could be solved as graph problems by ordering the events using domain-specific heuristics and then assign the events sequentially into valid time slots in such a way that no constraints are violated for each timeslot Carter.M.W(1986).(Sequential Methods)

• Timetabling problem is modeled as a set of variables (events) to which values (resources such as teachers and rooms) have to be assigned in order to satisfy a number of hard and soft constraints Brailsford.S.C et.al(1999). (constraint based methods).

• Timetabling problem is divided into a number of events sets. Each set is defined so that it satisfies all hard constraints. Then, the sets are assigned to real time slots to also satisfy the soft constraints as well. White.G.M(1979). (Cluster methods).

• Genetic algorithms (GAs), simulated annealing, tabu search, and other heuristic approaches apply nature-like processes to solutions or populations of solutions, in order to evolve them towards optimality. Abramson.D(1991), Hertz.T.A(1991), Paechter.T.B.et al. (1995). (Meta heuristic methods).

Since then, the literature has hosted a large number of papers presenting evolutionary methods and applications on such problems with significant success (Carter.M.W(1986)).

Since soft constraints satisfactory level fixing up optimality, mutation is done to achieve the same. With respect to the selection of soft constraint to be mutated/satisfied, different mutation strategies are proposed (Nandhini.M and Kanmani.S(2011)).

Diversity in population increases the global optimality. By proposing a combinatorial partially matched crossover operator, performance of diversity measure is increased by keeping the range of fitness unchanged/with less change (Nandhini.M and Kanmani.S(2011).

To get the global optimal as the result of genetic algorithm, diversity in population with proposed/used crossover and convergence with proposed mutation, could be combined. With these ideas, a combinational approach of proposed GA operators are tried in this work and its experimental results are compared with previous work.

This paper is organized as follows. Related works done on this domain is given in Section 2. Section 3, narrates the process of GA. In Section 4, usage of various operators of GA is

Tele:
E-mail addresses: mnandhini2005@yahoo.com,kanmani@pec.edu

described followed by simulation results are in section 5. Section 6 concludes with the future work.

Related Works

Researches are going on over past four decades on course timetabling problem. Domains like artificial intelligence, operations research applied to solve the problems. Researchers focused on usage of various algorithms, changes in operators and hybrid approaches. Due course of time, since GA helps in getting global optimal, it is used mostly to solve combinatorial problems. In general, works gone with GA has been divided into different categories: general GA architecture, changing operators in GA, GA with heuristics and combining with some other evolutionary algorithms viz., particle swarm optimization, Ant Colony optimization. Ali K. Kamrani, Ricardo Gonzalez (2008) developed genetic algorithm based solution approach to combinatorial optimization problems with depth first branch and bound algorithm and the local search with GA. Pongcharoen.P, Promtet.W, Yenradee.P, Hicks.C (2008) developed a tool SOTT by embedding genetic algorithms, simulated annealing and random search.

Salwani Abdullah and Hamza Turabieh (2008)]generated University Course Timetable Using Genetic Algorithms and Local Search with Tournament selection, single point crossover, random mutation, repairing and local search to improve the converging factor faster and had developed table with limited constraints. Wutthipong Chinnasri and Nidapan Sureerattanan(2010) 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.

Yu Zheng , Jing-fa Liu, Wue-hua Geng and Jing-yu Yang (2009), 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.

Salwani Abdullah and Edmund K.Burke and Barry McCollum (2007) described a hybrid evolutionary approach to the University Course Timetabling Problem. This mutation was done by selecting a course at random and making changes without violating feasibility. Evolution of population takes place over the repeated addition of GA operators, particularly selection, crossover and mutation and is shown in Datt.D, Deb.K and Fonseca.C.M(2007).

Yang, S. and Jat, S. N.(2010) designed Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling. It uses guided search strategy for storing information extracted from good individuals of previous generations and uses LS to improve the search efficiency.

Drifa Hadjidj and Habiba Drias.(2010) gave Grasp (Greedy Randomized Adaptive Search Procedure)and Guided Local Search for the examination timetabling problem and received better results.Mohamed Bader El Den and Riccardo Poli (2009) designed Grammar-based genetic programming framework via the evolution of constructive heuristics. It uses grammar to produce new generations based on graph coloring heuristics.

GA has been attempted so long as quoted in the related works. By taking detailed survey over past research work done on timetabling problem with GA, it could be identified the importance of soft constraints satisfaction in getting more optimal solution. Since no such strategies are there, we have proposed some mutation strategies (random selection, adaptive, goal directed with local search) with gene tuning approach, in which soft constraints are dealt in entirely different way (Nandhini.M and Kanmani.S(2011).

Consequently, crossover (uniform; combinatorial partially matching crossover) and selection (rank; grade) operators have proposed and applied without mutation to show its behaviour in the performance of diversity measures to get global optimum (Nandhini.M, Kanmani.S and Anandan. S(2011)) .With the aim of getting efficiency of both mutation and crossover, this paper is proposed with different combination of GA operators and those experimental results are compared with the related work found in the literature survey. Salwani Abdullah and Hamza Turabieh (2008).

Genetic Algorithm and Its Applications

Genetic algorithm is used to search large, nonlinear solution space where expert knowledge is lacking or difficult to encode. Moreover, it evolves from one population to another and produces multiple optima rather single local one. These characteristics make GA a suited tool for course timetable problem.

Evolutionary Process

The evolution process starts from a population of random individuals. It is known as a generation. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population based on their (fitness), modified (mutated or recombined) to form a new population, which becomes current population in the next iteration of the algorithm. GA is considered as one of the most powerful techniques in evolutionary algorithms, so GA has been employed as a tool that can handle a very complex search space with a high probability of success in finding the optimal solutions and flow diagram is given in Fig.1.

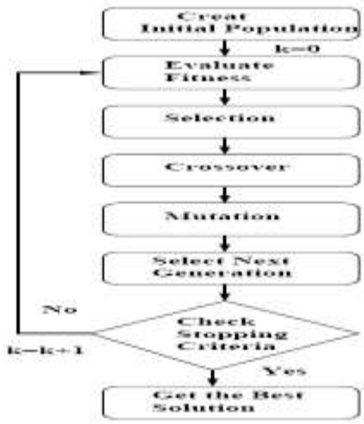

Fig 1. Evolutionary Process of General GA

Hybridization of genetic algorithms and SAHC with gene tuning

GA has been used for course timetabling problem and the proposed GA architecture is shown in a schematic diagram as in Fig.2. The parameters used in describing this problem are given in annexure-A. In the first phase of GA, random solution in the form of a conflict matrix is created with the dimensions of m by n where m denotes days in a week, n for timeslots of all classes.

Initial population (Chromosomes) Representation

The initial population consists of a number of chromosomes equal to the population size. Each chromosome is created using the constructive heuristic approach to avoid clashes and is represented as a three-dimensional matrix. Lower index represents periods, middle represents a day and upper represents a class. Then, the value of each cell (timeslot) of the matrix represents allotment scheduled in the corresponding class and period. The initialization procedure in Fig. 2 encodes the input data into chromosome representation. The result of initialization

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

Fig.2 Population Initialization Procedure

Genetic Operation

The schematic diagram of entire proposed process is shown in Fig.3. where proposed areas in GA are given in the shaded form.

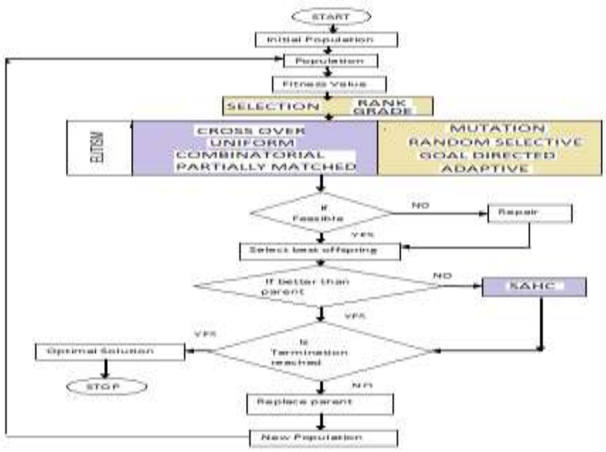

Fig 3. Schematic diagram of GA and local search algorithm

Fitness function

A factor to evaluate the timetable for finding its level of optimality is fitness function. This is calculated from the penalty cost and the validity of soft constraints. A chromosome that has 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 Constraints

If $j \in SC$ on *T* is satisfied, then *v(j)* =0, Otherwise *v*(*j*) = 1.

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 losing the best found solution. Here, 10% of chromosomes having higher fitness values are copied into new population in order to retain the best solution in the next generation.

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.

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.

Grade selection

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

The Procedure for Grade Selection is as follows:

Find the standard deviation (SD) for the individuals in the mating pool. 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) where $\overline{x}$: average cost of fitness , as average grade, and form range of higher grades by adding SD with average and form lower grades by deducting SD from 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 grades (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.

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 (Sivanandham. S.N and Deepa. S.N (2008) the reordering operators proposed is Combinatorial Partially Matched Crossover (PMX) and Uniform Crossover (UX).

Crossover rate

According to a user-definable crossover probability, crossover occurs during evolution and taken as 0.8 which is been taken form literature.

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 genes from their parents.

PMX proceeds as follows:

- The two chromosomes are aligned.
- Two crossing sites are selected uniformly at random along the strings, defining a matching section.
- The matching section is used to affect a cross through position by position exchange operation.
- Genes are moved to their new positions in the offspring.

Combinatorial Partially Matched Crossover (Combinatorial PMX)

Due to the sensitivity of this problem, altering timeslots within chromosomes result with uncertainty. Because of resource interrelated activities, crossover against two individuals' matching section could not be done in most of the times. This led us to propose a crossover operator, variant of PMX named as Combinatorial Partially Matched crossover.

Wherein possible combinations of matching section timeslots are formed by rotating the 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 timeslot exchanges is considered for further operations. With respect to this problem, mating is done between respective classes of two chromosomes. 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 Fig.4.

Uniform crossover

Uniform crossover (Sivanandham. S.N and Deepa. S.N(2008) 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.

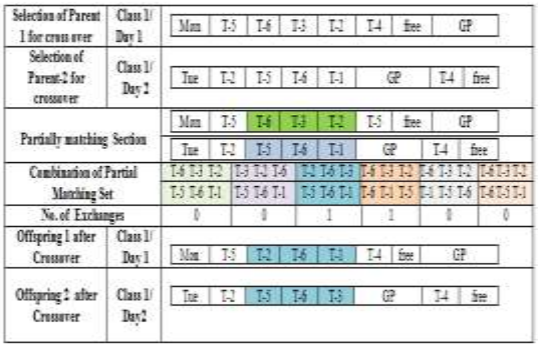

Fig.4 Procedure of Combinatorial PMX

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.

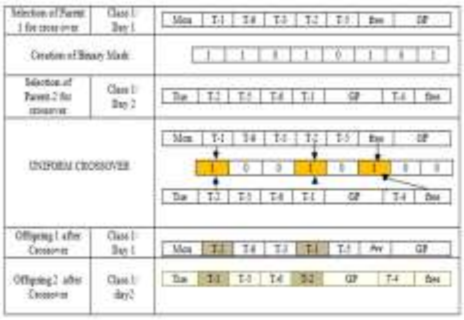

Fig.5 Procedure of Uniform Crossover

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 Fig.5. Implementation of these crossover operators might result with infeasible chromosomes and are recovered by repairing process.

Mutation

Mutation means randomly deriving change to the gene sequence of the chromosomes. In GA, mutation is a purely random operator, in which the probability that a gene will mutate is of low value at the time of initialization.

Gene Tuning

To make changes in nature of chromosomes, random changes has been done only on one timeslot so far in the literature ,which may not be giving guaranteed improvement in the chromosome. Optimality factor depends on level of soft constraints satisfaction. To improve convergence factor, soft constraint satisfaction has been proposed in the form of mutation. Soft constraints are made to satisfy during mutation, by fine tuning the timeslots (genes) which give violations. Hence named and proposed as gene tuning. Tuning some genes may affect the feasibility of timetable. Those timetables are fully recovered by applying repairing process in our work.

Selection of soft constraints led us the development of mutation strategies. First one is giving importance to all soft constraints. Second case is selecting a soft constraint giving better fitness value if satisfied. Satisfying soft constraint raising more penalty cost (more violation) is the reason for third case.

These three mutation strategies have been implemented in the name of Random selection; Adaptive; Goal Directed. The procedures for implementing all these soft constraint oriented mutations are described below. The performance of mutation strategies are tested with same data set and different population sizes.

To have improvement over the previous generation, the fixed number of chromosomes have to be mutated by selecting them from the mating pool of M solutions. Here, 3% of population is mutated in each generation which is taken from the literature.

Random Selection Strategy

With fixed rate, in one generation, mutation of one soft constraint is randomly selected and is been performed over the selected individual. Diagrammatical representation is shown in Fig.6.

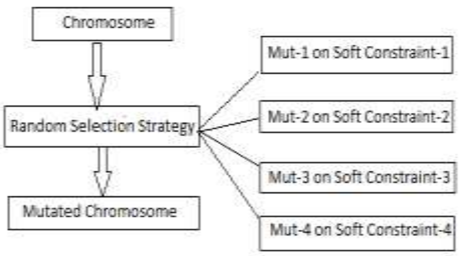

Fig.6 Block diagram of Random Selection Mutation Strategy

Adaptive Strategy

In one generation, selected individual undergoes mutation of all soft constraints and their corresponding fitness scores are calculated.

To have minimum fitness, mutated individual with the least score is adapted and is compared with its parent and if better, the parent gets replaced and is shown in Fig.7.

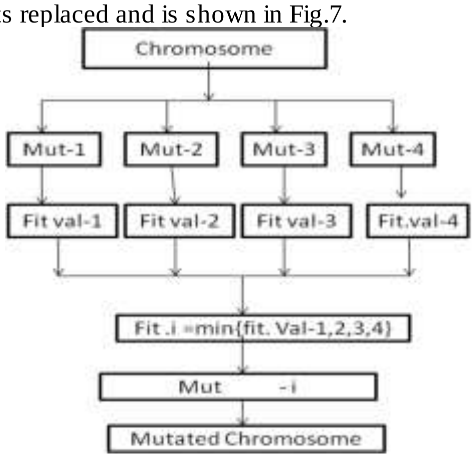

Fig.7 Block diagram of Adaptive Mutation Strategy

Goal Directed Strategy

In the selected chromosome, mutation of the soft Constraint that is violated more is performed in order to reach the goal of minimizing fitness and it is shown in Fig.8.

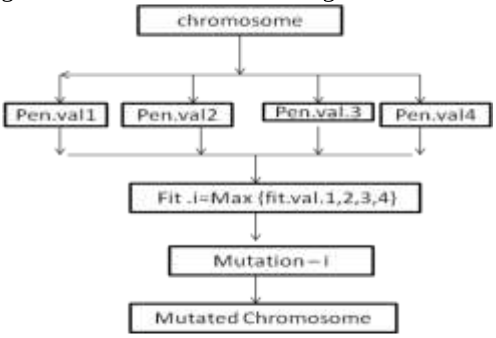

Fig.8 Block diagram of Goal Directed Mutation Strategy

Optimization with better convergence rate factor heuristics framed in the form of mutation. In the chromosomes, convergence is decided by soft constraint satisfaction. Proposed mutation strategies help in this regard. In the random selection, soft constraint is selected randomly, no control is over there. But in the other two, selection of soft constraints is decided based on the fitness value and could take chromosome to the better fitness.

Repair

Repairing is mainly done for removing the violation of hard constraints after reproduction operation. This function has composed of two distinct tasks: fault detection and fault correction. Knowing the location of the offending timeslots, repairing replaces these timeslots with free slots at first. If conflict raises, iteratively replaces with other timeslots entries in order to get rid of hard constraints violation (Nandhini.M, Kanmani.S and Anandan. S(2011)) .

Local Search -Steepest Ascent Hill Climbing (SAHC)

Inserting local search within genetic algorithm is considered as an effective way to produce high quality solution than using genetic algorithms alone Gani.T.A,Khader.T.A and Budiarto.R (2004), Nabeel R. and AL-Milli(2010) & Shengxiang yang and Sadaf naseem jat (2010). Especially during divergence in the population, we applied steepest ascent hill climbing algorithm steepest hill climbing to improve the timetable by reducing the number of soft constraint violations. In SAHC, all successors are compared and the closest to the solution is chosen. It tries all possible extensions of the current path instead of only one. By hybridizing SAHC with GA, in each generation to improve the feasibility of the individual, convergence rate would be very high and the optimal solution could be obtained by exploring to the maximum and covering the minimum distance. Performance of SAHC on CTP was done and implemented results are analyzed over various parameters [26]. Combining SAHC with mutation (Nandhini.M and Kanmani.S (2011)) and crossover (Nandhini.M , Kanmani.S and S. Anandan(2011)) operators also implemented. The side effect of SAHC in improving fitness is shown in the experimental results.

Termination Criteria

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

Simulation Results

The proposed algorithm has been implemented using Java (Jdk 1.6). Table. 1 shows the parameter for the genetic algorithm which are obtained from the literature (Salwani Abdullah and Hamza Turabieh (2008)).

Experiments on population sizes of 100, 200 400 and 600 for different generations with different parameters such as low fitness value and high fitness value after and before applying genetic operators have been done. In some cases, result reaches to stable state after several iterations, so we stop the algorithm if there is no improvement on the timetable.

To identify the performance of combination of genetic operators which we have proposed/used, the program is executed with different combination of operators given in Table.2

The results obtained for the above combinations of operators are given in the annexure-B. To analyze the Performance of the results, approach in the literature using genetic algorithms and local search by Salwani Abdullah and Hamza Turabieh[2008] also implemented and compared.

Comparison is presented as charts with fitness value in X-axis and generation in Y-axis. Each horizontal line consists of four sizes of population's (100, 200,400 &600) with lowest or highest fitness value in different generations range of fitness, grade , UX with goal directed prodcuing next better results.

In general from the figures .8,9,10&11, it is concluded that adaptive mutation strategy with selection and crossover operators combination is performing better by producing consistent result.

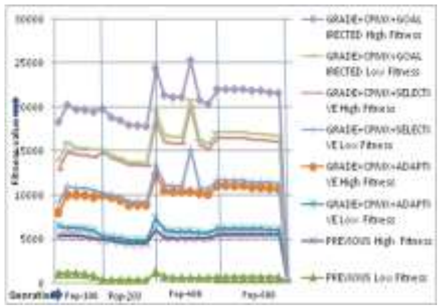

Fig 8. Performance comparison of Grade, CPMX with Mutation Strategies

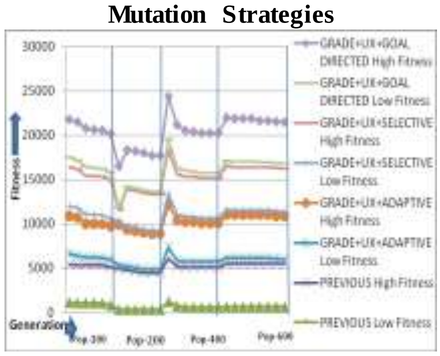

Fig 9 Performance comparison of Grade, UX with Mutation Strategies

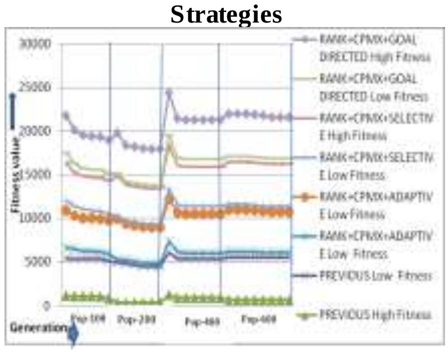

Fig 10 Performance comparison of Rank, CPMX with Mutation Strategies

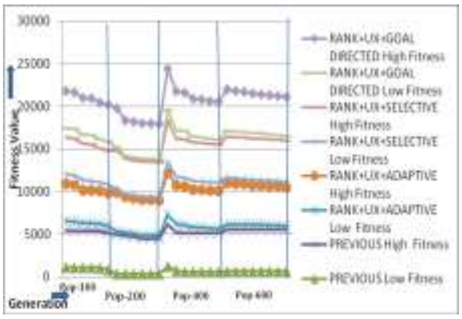

Fig 11 Performance comparison of Rank UX with Mutation Strategies

Following this, goal directed mutation strategy combined with Selection and crossover operators produces better quality results. From the above discussion, it is identified that combination of operators with adaptive mutation is producing most promising result in all the cases of small and large sizes of data set and is showing consistent better performance. Hence, it is declared that adaptive mutation is performing better than other mutation strategies and literature approach. Next to that, combination with goal directed mutation is performing better.

To analyze the effect of selection and crossover operators, outcome of combinations of adaptive mutation startegy, rank/grade selection and uniform/CPMX crossover are compared. It is found from the reduction in fitness that grade with CPMX combination producing better outcomes for some data sets sizes 100, 200. Rank with UX combination also producing better results for data set of size 600. For data set of size 400, rank, grade and UX/CPMX operators producing same results. From the observartions, it is concluded that grade with CPMX operators producing better results.

In order to show the combination of genetic operators which is possibly giving better results, comparison has been made among combination of operators with adaptive mutation. It is shown in the Fig.12 and found that combination of Grade selection, Combinatorial Partially Matching Crossover and Adaptive mutation strategy is giving more promising result than other three combinations and literature approach.

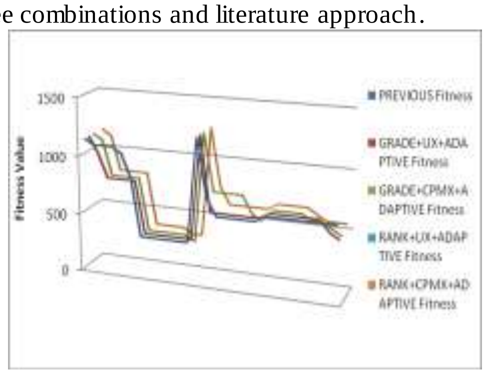

Fig 12. Performance comparison of Adaptive Mutation with Grade, Rank Selections and CPMX, Uniform crossover

Relative convergence rate for all combination of operators with adaptive mutation for population sizes viz. 100,200,400 & 600 have been found. It is identified from Fig.13 that Grade, CPMX and Adaptive mutation combination is yielding 31% of average rate of convergence which is better than all the others. If generation increases, there might be a chance of getting more near optimal solutions.

Hence, the potentiality of our genetic algorithm is proved. Main problem of GA that is time constraints could get relaxed with these operators. Sample course timetabling and laboratory timetabling is given in the annexure-C.

Conclusion

This paper framed with enhancement of operators in GA and local search algorithm. Combination of all operators giving good solution. To have random and liable individuals for genetic operation, grade selection is proposed. New Crossover operator is introduced to have diversity in population by having maximum possible timeslot exchanges during cross over. New strategies have been proposed for mutation in order to satisfy the soft constraints and to increase the convergence rate of process. A repair function is also presented that is totally able to change infeasible timetable to a feasible one. Steepest ascent hill climbing local search is applied to improve fitness value if found not better. These combinations produce more promising results. As future work, all proposed genetic operators might be implemented and tested over similar nature of problems.

References

1. Abramson. D, Constructing school timetables using simulated annealing: sequential and parallel algorithms, Management Science 37 (1991) 98–113.

2. Adamidis.P, Arapakis.P, Evolutionary algorithms in lecture timetabling, In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC 99),1999, pp.1145–1151.

3. Ali K. Kamrani, Ricardo Gonzalez,Genetic algorithms-based solution approach to combinatorial optimization problems, Inderscience, International Journal of Knowledge Management Studies , 2 (4)(2008)499 - 518.

4. Brailsford.S.C, C. N. Potts, B. M. Smith, Constraint satisfaction problems: Algorithms and applications, European Journal of Operational Research. 119 (1999) 557–581.

5. Burke.E.K,Newall.J.P, A new adaptive heuristic framework for examination timetabling problems, Technical Report NOTTCSTR- 2002, University of Nottingham, UK, School of Computer Science & IT, 2002,201-204.

6. Carter.M.W, A survey of practical applications of examination timetabling algorithms, Operations Research, 34 (1986) 193–202.

7. Carter.M.W,Laporte.G, Recent developments inpractical course timetabling, In:2nd Int'l Conf. on the Practice and Theory of Automated Timetabling, Springer Lecture Notes in Computer Science Series, 1408, 1990,pp.3-19.

8. Datta, D., Deb, K,Fonseca, C.M. ,Multi-objective evolutionary algorithm for university class timetabling problem, In:Evolutionary Scheduling Book , Series of Studies in Computational Intelligence , Berlin: Springer,49,2007,pp.197-236.

9. Drifa Hadjidj, Habiba Drias,Grasp and Guided Local Search for the examination timetabling problem, International Journal of Artificial Intelligence and Soft Computing 2(1/2) (2010) 103 - 114.

10. Gani.T.A,Khader.T.A,Budiarto.R, Optimizing examination timetabling using a hybrid evolution strategies,In: Proceedings of the 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, 2004, pp.345 – 349.

11. Hertz.T.A,, Tabu search for large scale timetabling problems, European journal of operations research 54 (1991) 39–47.

12. Mohamed Bader El Den, Riccardo Poli, Grammar-Based Genetic Programming for Timetabling, In: Proceedings of IEEE Congress on Evolutionary Computation, 2009, pp.2532-2539.

13. Nabeel R. AL-Milli ,Hybrid Genetic Algorithms with Great Deluge For Course Timetabling IJCSNS International Journal of Computer Science and Network Security, 10 (4) (2010).

14. Nandhini.M, Kanmani.S, Anandan.S, Performance Analysis Of Diversity Measure With Crossover Operators In Genetic Algorithm, submitted to WorldSciNet-International Journal of Computational Methods (IJCM) , Jan 2011.
15. Nandhini.M, Kanmani.S, Design of Gamut-LSsahc – A solver for course timetabling problems, Special issue of InderScience,Int. J. of Mathematical and Operational research, 2011, In press.
16. Nandhini.M, Kanmani.S, Design of Mathematical representation and optimization of course timetabling using local search, Int. J of Combinatorial Optimization and Informatics, 1(2) (2010),10-19.
17. Paechter.T.B, Norman.M.G, Luchian.H, Extensions to a memetic timetabling system, in: Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling, Lawrence Erlbaum Associates, Springer-Verlag LNCS 1153,1995,pp.251-265.
18. Pongcharoen.P, Promtet. W, Yenradee. P,Hicks.C.,Stochastic Optimization Timetabling Tool for university Course scheduling, International Journal of Production Economics, 112(2) (2008) 903-918.
19. Salwani Abdullah, Edmund K.Burke, and Barry McCollum,A hybrid evolutionary approach to the University Course Timetabling Problem, In:Proceedings of IEEE Congress on Evolutionary Computation 2007, pp.1764-1768.
20. Salwani Abdullah, Hamza Turabieh , Generating University Course Timetable Using Genetic Algorithms and Local Search ,In. Proceedings of Third 2008 International Conference on Convergence and Hybrid Information Technology (2008) pp.254-260.
21. Schaerf.T.A, A survey of automated timetabling, Artificial Intelligence Review 13 (2) (1993) 87–127.
22. Shengxiang yang,Sadaf naseem jat, Genetic algorithms with guided and local search strategies for university course timetabling,IEEE Transactions on systems, man, and cybernetics,(2010), pp.20-26.
23. Sivanandham. S.N ,S.N. Deepa,. Introduction to Genetic Algorithm, Springer (2008), ISBN: 9783540731894.
24. Tripathy.T.A, A lagrangian relaxation approach to course timetabling, Journal of the Operational Research Society 31 (1980) 599-603.
25. White.G.M,Chan.P.W, Towards the construction of optimal examination timetables, INFOR 17, (1979) 219–229.
26. Wutthipong Chinnasri, Nidapan Sureerattanan,Comparison of performance between different selection strategies on genetic algorithm with course timetabling problem', IEEE International Conference on Advanced Management Science (ICAMS) (2010) pp. 105-108.
27. Yang, S. Jat, S. N,Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling, IEEE Transactions on Systems, Man, and Cybernetics, Applications and Reviews,.99,(2010),1-14.
28. Yu Zheng Jing-fa Liu Wue-hua Geng Jing-yu Yang,Quantum-Inspired Genetic Evolutionary Algorithm for Course Timetabling, WGEC '09. 3rd International Conference on Genetic and Evolutionary Computing, (2009), 750-753.

Appendix – 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 an 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.

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.

Room Conflicts

• Laboratory periods for different classes assigned in a physical laboratory location must not overlap.

• Laboratory periods should come in the continuous timeslot either in the morning or in the evening session but not in the first period of both sessions.

Soft constraints

• At least one period gap might be given between the lecture periods of a teacher in a day.

• In adjacent days, two same periods should not have the same subject.

• First period of a day should be different from other day.

• Each staff should be given first period at least once in a week.

• Free periods should 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.

Fig. 13. Relative Convergence Rate

Pop. Size	PREVIOUS	GRADE+UX+ADAPTIVE	GRADE+CPMX+ADAPTIVE	RANK+UX+ADAPTIVE	RANK+CPMX+ADAPTIVE
100	0.26	0.30	0.30	0.26	0.30
200	0	0	0.14	0.14	0.14
400	0.52	0.52	0.56	0.56	0.52
600	0	0.23	0.23	0.23	0.15
Average Rate	20%	26%	31%	30%	28%

Table: 1 Parameter Setting For Genetic Algorithms

Parameter	Value
Generation Number	100,200,400,600
Population Size	100,200,400,600
Elitism	10%
Crossover Rate	80%
Mutation Rate	3%
Selection	Rank, Grade
Crossover Type	Uniform, Combinatorial partially matched
Mutation Type	Random selection, Adaptive, Goal directed

Table.2. Proposed Combination of Operators

Selection	Crossover	Mutation
Rank	UX	Random Selection
Rank	UX	Adaptive
Rank	UX	Goal Directed
Rank	PMX	Random Selection
Rank	PMX	Adaptive
Rank	PMX	Goal Directed
Grade	UX	Random Selection
Grade	UX	Adaptive
Grade	UX	Goal Directed
Grade	PMX	Random Selection
Grade	PMX	Adaptive
Grade	PMX	Goal Directed

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

Appendix - B

Table: 4 Performance of Adaptive Mutation with different Selection and Crossover operators

Population	Generation	Previous		Grade+ Ux+ Adaptive		Grade+ Pmx+ Adaptive		Rank+ Ux+ Adaptive		Rank +pmx+ Adaptive	
		Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low fitness	High fitness
100	1	1150	4300	1150	4300	1150	4300	1150	4300	1150	4300
	10	1100	4300	1000	4300	1100	1500	1100	4300	1100	3800
	25	1100	4300	800	3800	800	3800	950	3800	800	3800
	50	1100	4300	800	3800	800	3800	950	3800	800	3800
	75	1050	4300	800	3800	800	3800	900	3800	800	3800
	100	850	4300	800	3800	800	3800	850	3800	800	3800
200	1	350	4600	350	4600	350	4600	350	4600	350	4600
	50	350	4500	350	4150	350	4500	350	4150	350	4150
	100	350	4350	350	4150	350	4350	350	4150	350	4150
	125	350	4150	350	4150	350	4050	350	4150	350	4150
	150	350	4150	350	4000	350	4050	350	4150	300	4150
	200	350	4150	350	4000	300	4050	300	4150	300	4150
400	1	1250	4850	1250	4850	1250	4850	1250	4850	1250	4850
	100	750	4500	600	4500	750	4500	1000	4500	750	4500
	200	600	4500	600	4500	750	4500	1000	4500	600	4500
	250	600	4500	600	4500	750	4500	750	4400	600	4500
	300	600	4500	600	4400	750	4500	700	4400	600	4500
	350	600	4500	600	4400	600	4500	600	4400	600	4500
	400	600	4500	600	4400	550	4400	550	4400	600	4500
600	1	650	4850	650	4850	650	4850	650	4850	650	4850
	100	650	4850	650	4850	650	4850	650	4750	650	4850
	200	650	4850	650	4850	650	4850	650	4700	650	4850
	300	650	4850	650	4850	650	4850	650	4600	650	4750
	400	650	4850	650	4850	600	4750	650	4550	600	4700
	500	650	4850	650	4850	600	4750	600	4550	550	4700
	550	650	4850	550	4850	550	4700	550	4550	550	4700
	600	650	4850	500	4850	500	4700	500	4500	550	4700

Table .5 Performance of Grade Selection, CPMX crossover with various mutation strategies

POPULATION	GENERATION	PREVIOUS		GRADE+ CPMX+ ADAPTIVE		GRADE+ CPMX+ SELECTIVE		GRADE+ CPMX+GOALIRECTED	
		Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness
100	1	1150	4300	1150	4300	1150	4300	1150	4300
	10	1100	4300	1100	1500	1100	3800	1100	4300
	25	1100	4300	800	3800	1050	3800	1100	4300
	50	1100	4300	800	3800	800	3800	800	4300
	75	1050	4300	800	3800	800	3800	800	4300
	100	850	4300	800	3800	800	3800	800	4300
200	1	350	4600	350	4600	350	4600	350	4600
	50	350	4500	350	4500	350	4150	350	4250
	100	350	4350	350	4350	350	4150	350	4250
	125	350	4150	350	4050	350	4150	350	4200
	150	350	4150	350	4050	350	4150	350	4150
	200	350	4150	300	4050	350	4150	350	4100
400	1	1250	4850	1250	4850	1250	4850	1250	4850
	100	750	4500	750	4500	750	4850	750	4500
	200	600	4500	750	4500	650	4850	750	4500
	250	600	4500	750	4500	650	4850	750	4500
	300	600	4500	750	4500	4850	4850	750	4500
	350	600	4500	600	4500	650	4850	600	4500
	400	600	4500	550	4400	650	4500	600	4500
600	1	650	4850	650	4850	650	4850	650	4850
	100	650	4850	650	4850	650	4850	650	4850
	200	650	4850	650	4850	650	4850	650	4850
	300	650	4850	650	4850	650	4850	650	4850
	400	650	4850	600	4750	650	4850	650	4850
	500	650	4850	600	4750	650	4850	650	4850
	550	650	4850	550	4700	650	4750	650	4850
	600	650	4850	500	4700	650	4750	650	4850

Table: 6 Performance of Grade Selection, Uniform crossover with various mutation strategies

POPULATION	GENERATION	PREVIOUS		GRADE+ UX+ ADAPTIVE		GRADE+ UX+ SELECTIVE		GRADE+ UX+GOAL DIRECTED	
		Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness
100	1	1150	4300	1150	4300	1150	4300	1150	4300
	10	1100	4300	1000	4300	1100	4300	1100	4300
	25	1100	4300	800	3800	1100	4300	1100	4300
	50	1100	4300	800	3800	1100	4300	950	4300
	75	1050	4300	800	3800	1050	4300	950	4300
	100	850	4300	800	3800	850	4300	950	4300
200	1	350	4600	350	4600	350	1300	350	4600
	50	350	4500	350	4150	350	4150	350	4150
	100	350	4350	350	4150	350	4150	350	4150
	125	350	4150	350	4150	350	4150	350	4150
	150	350	4150	350	4000	350	4150	350	4000
	200	350	4150	350	4000	350	4150	350	4000
400	1	1250	4850	1250	4850	1250	4850	1250	4850
	100	750	4500	600	4500	750	4500	750	4850
	200	600	4500	600	4500	600	4500	750	4500
	250	600	4500	600	4500	600	4500	600	4500
	300	600	4500	600	4400	600	4500	600	4450
	350	600	4500	600	4400	600	4500	600	4450
	400	600	4500	600	4400	600	4500	600	4450
600	1	650	4850	650	4850	650	4850	650	4850
	100	650	4850	650	4850	550	4850	650	4850
	200	650	4850	650	4850	550	4850	650	4850
	300	650	4850	650	4850	550	4850	650	4850
	400	650	4850	650	4850	550	4850	600	4650
	500	650	4850	650	4850	550	4850	600	4650
	550	650	4850	550	4850	550	4850	600	4650
	600	650	4850	500	4850	550	4850	600	4650

Table.7: Performance of Rank Selection, CPMX crossover with various mutation strategies

POPULATION	GENERATION	PREVIOUS		RANK+ CPMX+ ADAPTIVE		RANK+ CPMX+ SELECTIVE		RANK+ CPMX+ GOAL DIRECTED	
		Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness
100	1	1150	4300	1150	4300	1150	4300	1150	4300
	10	1100	4300	1100	3800	1100	3800	1100	3800
	25	1100	4300	800	3800	1050	3800	900	3800
	50	1100	4300	800	3800	900	3800	900	3800
	75	1050	4300	800	3800	900	3800	900	3800
	100	850	4300	800	3800	800	3800	800	3800
200	1	350	4600	350	4600	350	4600	350	4600
	50	350	4500	350	4150	350	4150	350	4150
	100	350	4350	350	4150	350	4150	350	4150
	125	350	4150	350	4150	350	4150	350	4150
	150	350	4150	300	4150	350	4150	350	4150
	200	350	4150	300	4150	350	4150	350	4150
400	1	1250	4850	1250	4850	1250	4850	1250	4850
	100	900	4500	750	4500	900	4500	900	4500
	200	900	4500	600	4500	900	4500	900	4500
	250	900	4500	600	4500	900	4500	900	4500
	300	900	4500	600	4500	900	4500	900	4500
	350	900	4500	600	4500	900	4500	900	4500
	400	900	4500	600	4500	900	4500	900	4500
600	1	650	4850	650	4850	650	4850	650	4850
	100	650	4850	650	4850	650	4850	650	4850
	200	650	4850	650	4850	650	4850	650	4850
	300	650	4850	650	4750	650	4850	650	4850
	400	650	4850	600	4700	650	4850	650	4850
	500	650	4850	550	4700	650	4850	600	4750
	550	650	4850	550	4700	650	4850	600	4750
	600	650	4850	550	4700	650	4850	600	4750

Table: 8 Performance of Rank Selection, Uniform crossover with various mutation strategies

POPULATION	GENERATION	PREVIOUS		RANK+ UX+ ADAPTIVE		RANK+ UX+ SELECTIVE		RANK+ UX+GOAL DIRECTED	
		Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitness
	1	1150	4300	1150	4300	1150	4300	1150	4300
	10	1100	4300	1100	4300	1150	4300	1100	4300
	25	1100	4300	950	3800	1150	4300	1100	4300
	50	1100	4300	950	3800	1100	4300	1100	4300
	75	1050	4300	900	3800	1000	4000	1100	4300
100	100	850	4300	850	3800	1000	4000	1100	4300
	1	350	4600	350	4600	350	4600	350	4600
	50	350	4500	350	4150	350	4150	350	4150
	100	350	4350	350	4150	350	4150	350	4150
	125	350	4150	350	4150	350	4150	350	4150
	150	350	4150	350	4150	350	4150	350	4150
200	200	350	4150	300	4150	350	4150	350	4150
	1	1250	4850	1250	4850	1250	4850	1250	4850
	100	750	4500	1000	4500	1000	4500	1000	4500
	200	600	4500	1000	4500	1000	4500	1000	4500
	250	600	4500	750	4400	1000	4500	750	4400
	300	600	4500	700	4400	1000	4500	700	4400
	350	600	4500	600	4400	1000	4500	600	4400
400	400	600	4500	550	4400	1000	4500	600	4400
	1	650	4850	650	4850	650	4850	650	4850
	100	650	4850	650	4750	650	4850	650	4750
	200	650	4850	650	4700	650	4850	650	4700
	300	650	4850	650	4600	650	4850	650	4600
	400	650	4850	650	4550	650	4850	650	4550
	500	650	4850	600	4550	650	4850	600	4550
	550	650	4850	550	4550	650	4850	550	4550
600	600	650	4850	500	4500	650	4850	550	4550