EVALUATION OF MUTATION EFFECT IN MEMETIC ALGORITHM

Nikita Kumari and Dr. Mukesh Choudhary

Department of MCA, Jodhpur National University, Jodhpur

Abstract: Memetic algorithm generates solutions to optimization problems using techniques inspired by natural evolution, like crossover, selection and mutation of genetic algorithm and local search. Our motivation is the application of evolutionary algorithms for solving real-world optimization problems. In this work ,a new concept called gene tuning associates with mutation for solving the multi objective soft constrained combinatorial problems is introduced. Also, various strategies of mutation regarding the selection of soft constraints are introduced and investigated on course timetabling problem. The experimental results give an indication towards promising mutation for practical application.

Keywords: Memetic algorithm, genetic algorithm, local search, multi objective, soft constrained, combinatorial problems

1. Introduction

A combinatorial optimization program, in which several objectives are considered, is called a *multi objective combinatorial optimization* problem, or *MOCO* problem. The decision making associated with the scheduling problem belongs to the category of combinatorial problems. Scheduling problems like timetabling, job scheduling etc., are the process of generating the schedule with multiple objectives, follows MOCO problem.

Reasons for scheduling complexity include [1] : Scheduling is a feasibility problem. The final solution must accomplish all the problem constraints. Another objective to be satisfied is the optimization of an evaluation function, adjusting to certain criteria as cost, lateness, process time, inventory time, etc., Some scheduling problems have many constraints due to the unavailability of resources, due dates, etc., Constraint representation cannot express the importance of the value domains Since exact approaches are inadequate and requires too much computation time on large real world scheduling problems, heuristics and meta heuristics are commonly used in practice.

Genetic algorithms(GA), Simulated annealing(SA), Tabu search(TS), Artificial neural networks(ANN), Greedy randomized adaptive search procedures(GRASP), Threshold algorithms, Scatter, Variable neighborhood search(VNS), Cooperative search systems are some of the meta heuristics applied for optimization of combinatorial problem over 20 to 30 years[2]. GA helps in diversifying the search in order to get global optima and local search techniques help in intensifying the search. This motivated us to form a hybrid model named as memetic algorithm [3],[4] by combining GA with local search to get the exertion of both.

In the past years, evolution of the population in GA took place with the vast introduction of GA operators particularly on selection, crossover and mutation [5,6,7] and variety of representation of chromosomes [8, 9] on various applications like travelling salesman problem, multi job shop scheduling, timetabling etc. Although the existing algorithms in the literature can increase the convergence rate and search capability of the simple genetic algorithm to some extent, the mutation operators used in these algorithms have not sufficiently made to use the characteristics of the problem structure. Most of genetic operators only change the form of encoding and are difficult to integrate the merit of the parent individuals. Sufficient use of information in the problem structure and the inspiration of soft constraints satisfaction which decide the optimality factor, motivated us to introduce a concept a called gene tuning and to propose domain specific reproduction strategies with mutation like *Random Selection, Adaptive mutation and Goal Directed* to generate the optimal solution with minimum generation.

To balance the exploration and exploitation abilities, local search (SAHC) is applied simultaneously with customized GA architecture. This hybrid combination called memetic has been tried over course timetabling problem (CCTP) and its versatility is proved with more promising results. The organization of the rest of the paper is as follows. The discussion on evolution of proposed model is given in Section 2. In Section 3, the problem definition of CCTP with its constraints and chromosome representation are mentioned. In Section 4, proposed methodology with design of GA operators is described and data set is in Section 5. The evaluation of this model is given in section 6 followed by conclusion in Section 7.

2. Motivation

In the past, promising combinations of problems could be further investigated until an adequate solution was found. These traditional methods of solution could easily miss a potentially good solution since combinatorial problems tend to involve very large search spaces. In recent decades, researchers have developed evolutionary algorithms to solve the combinatorial problems with practical sizes. Evolution drives the population to better and better individuals. GA suggested by Holland [10] is an evolutionary algorithm based on the principles of survival of the fittest and adaptation originated from Darwin's evolution theorem. This algorithm increases the fitness of solutions by the simulation of evolution process as Goldberg described [9]. Many GAs have been applied to solve different *np*-hard combinatorial optimization problems like CCTP [11,12] etc.,. Olivia Rossi-Doria and Ben Paechter[3] proposed an memetic algorithm for university course timetabling and its performance was proved with encouraging results. Pongcharoen et al. [13] developed a tool called SOTT by embedding genetic algorithms, simulated annealing and random search for solving combinatorial problems.

Chinnasri and Sureerattanan [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. Moreover, an important issue in combinatorial problems is handling the constraints. Constraints are duly framed for satisfying and optimizing the problem entities' objectives. Often, such objectives are in conflict with each other. In the work of Salwani Abdhulla et. al.[15] during GA process with tournament selection, single point crossover, random mutation and repairing, the offspring fitness was improved by the local search algorithm on university course timetabling problem(UCTP). This idea became the stepping stone for presenting the same in the form of mutation. Since our problems are more domain specific and soft constrained, by exploiting that concept , mutation has been proposed with different strategies in this work.

To improve the outcome of GA operators, proposed to apply SAHC local search. To lead out the searching direction through the best, instead of random local search, the best among the neighborhood space is taken by SAHC. In this paper, it is showed that this bio-inspired algorithm with local search named memetic could be used for solving multi constrained and multi objective combinatorial problems. The proposed model is described in the following section.

3. College Course Timetabling Problem

The problem involves assigning lecture activities to timeslots in the lecture hall and laboratories subject to laborious hard and soft constraints. Hard constraints must not be violated, but the violation of soft constraints should be reduced. A timetable that satisfies the hard constraints is known as a feasible solution. The following section discusses the timetabling of the Bachelor of Technology course offered in the Department of Information Technology, Pondicherry Engineering College.

3.1 Problem description

This course contains four classes (each for an year of study). The framework of each course in the institute is of the form 5 (days) x 8 (periods). Timeslots represents intersection of day and period. In each day, morning and afternoon session has four periods. Each semester has six theory and three laboratory subjects. Each theory should be allotted for four periods per week and laboratory subject for three continuous periods in a week. Due to room conflict, each practical will be conducted for 3 days by dividing students into three batches. Thereby, each practical should be monitored by a staff for nine periods. Co-curricular activities such as placement and training/clubs and societies for three periods and seminar/group discussion for two periods must also be allotted for each class. The parameters specified in CCTP is given in Table .1.

Table.1 Parameters Specification

Sl. No.	Description	Total periods
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 / club and societies	3
10	No. of seminar/ group discussion periods	2
11	No. of free periods	2
12	Total hours per week (including free periods)	40

3.2 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 teachers 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

3.3 *Soft constraints*

- At least one period gap should 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
- Maximum of two theories/one theory and one lab/two theories and one practical in a day only can be scheduled for a teacher in a day

3.4 Objective function

The fact that strength of soft constraint satisfaction decides the optimality level makes us to propose objective function with the penalty cost and validity of soft constraints. To exhibit the degree of violation of soft constraints, penalty costs will be assigned. The lesser violation of soft constraints will give the minimum fitness value. So that the objective function is defined as,

$$\text{Min} \quad f\,(\text{T}) = \sum_{j=1}^{n} P_j x V_j$$

Where,

P_j: Penalty cost of Soft Constraint(SC) j on T

V_j: Validity of Soft constraint j

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

3.5 Chromosome representation

Chromosome (timetable) is represented in 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. *N* indicates the number of such chromosomes used to form the population. Institution specific is its main trait. All timeslots are to be filled with events. For each laboratory periods, students are grouped into batches and allotted with different laboratories. One teacher should be assigned for each group of students in the timetable provided laboratory and teachers conflict must be satisfied.

4. Design of Proposed Methodology

To design a robust steady state GA, one needs to employ the following two stages: designing different operators to create the algorithm; and tuning the GA parameters.

4.1 *Elitism*

*The r*eproduction which is done on random selection of chromosomes might destroy present individuals which had the best fitness. At least 10 % of the next population has to be retained from the current population, so that the core of good solutions is maintained. All the other chromosomes are created with other genetic operations.

4.2 Rank Selection

Selection is the process of choosing parents from the generated population to undergo genetic operations like mutation or crossover. Chromosome having minimum fitness is assigned with higher rank. The higher rank (worst) solutions are taken for improvement in the successive steps.

4.3 Uniform Crossover(UX)

Crossover encourages the exploration of search space.In UX[16], 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 UX procedure is explained on CCTP in Fig. 1 and Fig.2.

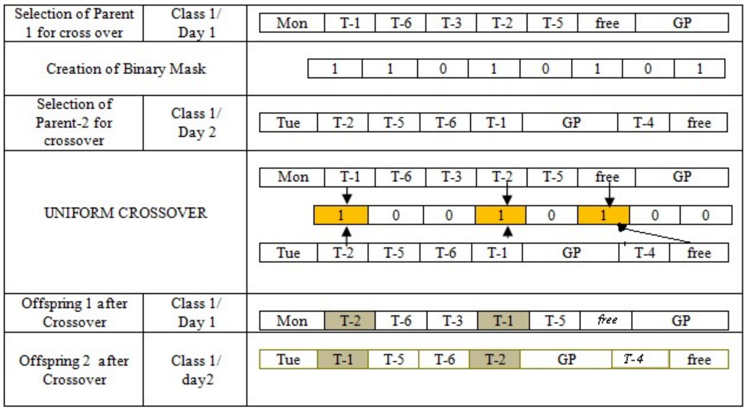
Selection of Parent 1 for cross over
Class 1/ Day 1
Mon | T-1 | T-6 | T-3 | T-2 | T-5 | free | GP
Creation of Binary Mask
1 | 1 | 0 | 1 | 0 | 1 | 0 | 1
Selection of Parent-2 for crossover
Class 1/ Day 2
Tue | T-2 | T-5 | T-6 | T-1 | GP | T-4 | free
UNIFORM CROSSOVER
Mon | T-1 | T-6 | T-3 | T-2 | T-5 | free | GP
1 | 0 | 0 | 1 | 0 | 1 | 0 | 0
Tue | T-2 | T-5 | T-6 | T-1 | GP | T-4 | free
Offspring 1 after Crossover
Class 1/ Day 1
Mon | T-2 | T-6 | T-3 | T-1 | T-5 | free | GP
Offspring 2 after Crossover
Class 1/ day2
Tue | T-1 | T-5 | T-6 | T-2 | GP | T-4 | free

Fig. 1 Procedure for UX on CCTP

```
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. 2: Uniform Crossover on CCTP

4.4 Mutation

In general the change in genes is done on random basis. From the past researches, it is inferred that there is no guarantee of improvement in the solution by these random changes.

4.4.1 Gene Tuning

To achieve this, removing the violation of soft constraints is attempted through mutation. This is achieved by tuning the genes and hence named as *gene tuning*. Side effect of tuning some genes resulting with infeasible schedules. To maintain feasibility, schedules affected by reproduction are fully recovered by repairing process.

4.4.2 Mutation Strategies

To identify the soft constraint to be satisfied by mutation, the following three strategies on mutation have been proposed.

- Mutation with random selection : Gene tuning is done on randomly selected soft constraint
- Mutation with adaptive strategy : Selecting the soft constraint resulting to minimum fitness score by gene tuning
- Mutation with goal directed : Among the existing soft constraints violation, the soft constraint raising more violation is corrected by gene tuning.

Optimization with fast convergence is the 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.

4.5 Repair

Repairing is mainly done for removing the violation of hard constraints after mutation operation. This function has composed of two distinct tasks: *fault detection* and *fault correction*. Knowing the location of the offending timeslots, repair process rectifies infeasible chromosomes. The GA in this design has repair processes that ensure all infeasible chromosomes are repaired. The repair function of this design is divided into two steps.

Step 1: Find the free timeslots for each class and replace the conflicting subject.

Step 2: If conflict continues, by finding the free timeslots in all the three classes repairing is done. If teacher conflict raises, replace free timeslot with other subject entries till receiving feasible timetable as output.

4.6 Local Search (Steepest Ascent Hill Climbing)

A very effective local search consisting of a stochastic process is used in three phases:

➢the first phase to improve an infeasible solution (timetable/jobs schedule) so that it becomes feasible; (Repair Function)

➢the second phase to increase the quality of a feasible solution by reducing the number of soft constraint violation (Mutation) and

➢the third phase is to improve the quality of the feasible solution by interchanging the genes.(SAHC)

The combination of GA operators with local search resulting to three memetic algorithms with enhanced GA and SAHC as shown in Table.2 and their performance are evaluated amongst them and against GA with LS by Abdhulla et.al.(2008).

Table.2 Proposed Memetic Algorithms

Memetic Algorithms	SELECTION	CROSS OVER	MUTATION STRATEGY	LOCAL SEARCH
MA1	RANK	UX	RANDOM SELECTION	SAHC
MA2	RANK	UX	ADAPTIVE	SAHC
MA3	RANK	UX	GOAL DIRECTED	SAHC

The probabilities of GA parameters which give better performance have been identified from various experiments considered and given in Table.3

Table.3. GA parameters

Parameter	Probability
Pop. Size	100,200,400, 600
Elitism	.1
Selection	Rank
Crossover	.8
Mutation	.03

5. Data Set

The instances of various sizes (SD(Small) ,MD(medium) and LD(Large)) of the actual scenario of timetables for the B.Tech course offered by Department of Information Technology at Pondicherry Engineering College for both odd and even semesters have been considered.The earlier work taken for comparison to prove the versatility of this proposed algorithm is GA with Local Search on UCTP by Salwani Abdhulla et.al .(2008).

6. Evaluation of Algorithms

This section describes the performance evaluation of the proposed memetic algorithms with the earlier works of GA with local search[15]. To analyze the performance, the lowest and highest fitness of population in different generations for various population sizes are tabulated in Table.5 for the algorithms MA1- MA3 and GA with Local Search. The quality of solution is measured with respect to the *relative fitness reduction* (RFR%) percentage over the number of repetitions (runs), which is calculated as follows.

$$RFR\% = \frac{\text{Best Fitness at initial generation} - \text{Best fitness at } n^{th} \text{ generation}}{\text{Best Fitness at initial generation}} * 100$$

To analyze the obtained data, the RFR% of various instances like SD1 (100), SD2 (200), MD1 (400) and LD1 (600) for different algorithms are tabulated in Table. 4. With respect to RFR%, higher the RFR%, gives more optimal solution.

Table.4. RFR% CCTP for different data sets

Data Set	**Pop. Size**	**GA with LS**	**MA1**	**MA2**	**MA3**
SD1	100	**26**	13	26	22
SD2	200	**0**	**0**	**14**	**0**
MD1	400	**52**	20	56	52
LD1	600	**15**	7	23	17
Average RFR%		***23***	***10***	***30***	***23***

6.1 Performance of GA with Local Search

To identify the cause of local search in improving optimality in the two problems, CCTP has been implemented on GA with local search approach as in [15]. From the responses of GA with LS on CCTP, for some data sets (SD2 : Population Size: 200) which are starting with low fitness value in its initial population, no improvement on fitness is found.(ie. RFR% = 0). Even then, it gives an average RFR% value as 23, which is better than some proposed algorithms. Hence, existence significant fitness reduction in this algorithm is found.

6.2 Analysis of proposed algorithms performance

The results of MA1-MA3 have been compared amongst and against GA with LS to identify the best algorithm. Analysis is done on two cases like RFR%, change in fitness while growth in generation over the data in the Tables.4 and 5.

Case 1: RFR%

Among three memetic algorithms, MA2(30) is giving the highest average RFR%, which means higher convergence towards optimal value. Invariably in all data sets , it is giving higher RFR%. Next to MA2, MA3(23) is giving better RFR%, but which is producing same average RFR% as GA with LS(23). MA1(10) is giving lower average RFR%, which should be the reason of randomly selected soft constraint satisfaction in mutation.

It is appreciatable that, MA2 is giving RFR% (14) for SD2. But ,there is no fitness reduction in the other algorithms. With these, discussions, it is concluded that adaptive mutaion is performing better than other proposed and exixting algorithms.

Case 2: Change in Fitness

The computational part of adaptive mutation is heavier than other two strategies due to its habitat and is obvious with its execution time. From Table.5 , it is observed that , in MA2 more fitness reduction is found in the earlier generation of the process. In some instances, there in no improvement in fitness in the later generation. Once it reaches some pleased solution, the solution gets stagnated for a period of time. The fitness change in MA3 considerably better than MA1 and the reduction happens throughout the entire process. This is the reason of gene changes during recombination operation. Hence, it is experimentally proved that mutation with adaptive strategy is better than other mutation strategies by producing consistent and promising results. Among other two, mutation with goal directed strategy is performing better than mutation with random selection.

Table.5 Performance of fitness change on CCTP using GA with LS ,MA1,MA2 and MA3

POPULATION	GENERATION	GA with Local Search		RANK+UX + ADAPTIVE(MA2)		RANK+UX+ SELECTIVE(MA1)		RANK+ UX+GOAL DIRECTED(MA3)	
		Low Fitness	High Fitne	Low Fitness	High Fitness	Low Fitness	High Fitness	Low Fitness	High Fitne
100	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	1000	4300
	100	850	4300	850	3800	1000	4000	900	4300
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	350	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	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	45	600	4400	1000	4500	600	44

			00						00
	400	600	4500	550	4400	1000	4500	600	4400
600	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	600	4850	600	4550	600	4850	600	4550
	550	600	4850	550	4550	600	4850	550	4550
	600	550	4850	500	4500	550	4850	550	4550

7. Conclusion

The proposed memetic algorithms with enhanced GA and SAHC produced promising results for all combinations of proposed operators. Also, its robustness is proved by obtaining better performance than algorithms taken from the literature. Since the operators have been designed in order to reduce the complexity of adjusting resources according to the constraints, these models definitely help in optimizing soft constrained combinatorial problem with minimum execution time.

References

[1] Garrido, A., Salido Barber, M. A., López, F. A. , "Heuristic methods for solving job-shop scheduling problems", *Workshop on New Results in Planning, Scheduling and Design*,pp.44-49,2000.

[2] Schaerf A., "A Survey of automated timetabling", *Artificial intelligence review*, 13, pp.87—127, 1999.

[3] Olivia Rossi-Doria,Ben Paechter, "A memetic algorithm for University Course Timetabling", *In Proc. Of Combinatorial Optimization(CO'2004)*,Scotland, 2004.

[4] Socha,K , "The influence of run time limits on choosing ant system parameters", In *Proc. Of Genetic and Evolution Conference, LNCS, verlag,* pp. 49-60, 2003.

[5] Datta, D., Deb, K., Fonseca, C. M. , "Multi-objective evolutionary algorithm for university class timetabling problem", *Series of Studies in Computational Intelligence*, Berlin: Springer, 49, pp.197-236, 2007.

[6] Ghaemi, S., Vakili, M. T., Aghagolzadeh,, " Using a genetic algorithm optimizer tool to solve university timetable scheduling problem", *International Symposium on Signal Processing and its Applications*, pp.1-4, 2007.

[7] Royachka, K., Karova, M, "High-performance optimization of genetic algorithms" *IEEE International Spring Seminar on Electronics Technology*,pp.395-400, 2006.

[8] Qi, J. G., Burns, G. R.,Harrison, D. K, “The application of parallel multi population genetic algorithms to dynamic job-shop scheduling”, *International Journal of Advanced Manufacturing Technology*, 16, pp.609-615, 2001.

[9] Goldberg, D. E, “*Genetic Algorithms in Search,* Optimization and Machine Learning. USA: Addison-Wesley, Boston, MS, 1989.

[10] Holland, J. H., “ *Adaptation in Natural and Artificial Systems*. USA: University of Michigan Press, Boston, MS, 1975.

[11] Ali K Kamrani,Ricardo Gonzalez.,“ Genetic algorithms-based solution approach to combinatorial optimisation problems”, Int. *J. of Knowledge Management Studies*, 2, 4, pp.499 – 518, 2008.

[12] Caldeira, J.Agostinho, C, “School timetabling using genetic search”, *Practice and Theory of Automated Timetabling*, pp.115–122, 1997.

[13] Pongcharoen,P., Promtet, W., Yenradee, P, Hicks,C., “ Stochastic Optimization Timetabling Tool for university Course scheduling”, *International Journal of Production Economics*, 112,2, pp.903-918,2008.

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

[15] Salwani Abdullah, Hamza Turabieh, “ Generating university course timetable using genetic algorithms and local search”, *IEEE Third International Conference on Convergence and Hybrid Information Technology*, pp.254-260, 2008.

[16] Sivanandham, S. N., Deepa, S. N., “ Introduction to Genetic Algorithm. New York: Springer Berlin Heidelberg, 2008.