Scientific Journal of Mechanical and Industrial Engineering

Sci. J. Mech. Ind. Eng., 2(4): 82-85 (2013)

University Examinations Scheduling Problem Using Genetic Algorithms

A. Goli[1*] , M. Aghaei[2]

[1]*Department of Industrial Engineering Isfahan University of Technology, Isfahan, Iran*

*Corresponding author's E-mail: a.goli@in.iut.ac.ir

Abstract

Exam scheduling at universities and institutes of higher learning is known as an issue with heavy constrain so that it can contract at issue NP-COMPLETE. In many cases it is necessary to manually create exams timetable that it require much time and lacks quality so that in many cases , student complain of inappropriate examination timetable. In this situation finding the best manually is not an easy task. Studies have shown the use of evolutionary algorithms, which offers a good response time may not be optimal but the constraints is necessary for use to consider and provide us an acceptable exam timetable. In this paper, with using genetic algorithms with a completely random population and with changes in normal process of this algorithm, we try to solve the population and provide the best exam timetable in a short time.

Original Article

Received 15 Dec. 2013
Accepted 30 Dec. 2013

Keywords:
Genetic Algorithm
Exam Scheduling

INTRODUCTION

The compilation of university exam timetable is a difficult task which educational institutes deal with it. Solve the problem of university exam timetable manually, needing extensive time and resources. Due to the complexity of solving the problem manually and automatically resolve schedule exams, many researchers have done a lot of research in this field. Multiple styles has been presented for this Field and in this direction for example stimulated annealing [5], constant logic programing [10], linear programing [8], graph coloring [14]. New research has been leaded toward evolutionary algorithms [11]. This algorithm has been known as a powerful technique in solving complicated problem of exam scheduling. We can imply to these cases [20] of the most important:

-Genetic algorithm
-colonial competitive algorithm
-Ant colony optimization
- Memetic algorithm

That among them genetic algorithm is conducted the best for solving problem of exam scheduling [4]. Exams scheduling is a subset of a larger problem calls scheduling that is assigning exams to limited number of times as all of important limitation are observed. There is some limitation which it isn't necessary to observe them but make the ultimate answer, desirable, for example: if there is a space between every two exams of a student more than one day .This is a sample of this kind of limitation. These limitations are called soft limitation.

Literature review

From the tasks has been done in this field, can imply to this case: Picher and Rankin represent a method based on Memetic algorithm with an effective local research which was used on first match timetable data in 1994 [12]. Brok, Elimen and Vier have suggested genetic algorithm in computer Sciences College of Nottingham University which is used only on acceptable time distances and is related mutually with user [13].

From a knowledgeable and intelligent genetic algorithm for designing exams timetables has been gain. Their view has been a two-step view that in first step difficult limits and in second step soft limits has been tested. In this research specially advantage of genetic algorithm has been cleared and reported hopeful results from that [9].

Problem of exams schedule and its limitations: Many of limitations of problem of exams timetable schedule from a university to another is different [6]. But two limitations are base of exams schedule:

To cite this paper: Goli A, Aghaei M. 2013. University examinations scheduling problem using genetic algorithms. *Sci. J. Mech. Ind. Eng.*, 2 (4): 82-85.
Journal homepages: http://sjmie.science-line.com/

-Any student shouldn't be present at one time in more than one place.

-Number of persons who participate in exam at the same time shouldn't be more than number of chairs.

Furthermore, users are interested in regarding below limitation in exams schedule table.

-Any student shouldn't have two exams in two nearly turns.

Another category of limitation is management limitation that professors and educational management of university tend to consider those:

X exam must conduct in special class;

X and Y exam must conduct at the same time or in the same day.

In many cases divide limitation in two kind of hard and soft [11, 20] and also with a very little chance, can find a suitable exams schedule table which has all of said limitation [6] for this reason we aren't satisfied with these two category in this paper, and the limitations categorize based on importance and when any limitation isn't considered a special fine is deter mind is determined for its amount of hardness or softness by application factor. In another expression for less important limitation smaller fine factor and for more important limitation, greater fine factor is accounted.

Exams schedule problem with genetic algorithm

-The exams must be conducted in determined number of day, from before.

-The number of classes and capacity of every class for conducting exams must be fixed to the end of exams.

The professors of every Course at limit hours can present in exam's session and absence of related professor for that exam isn't desirable and is a soft limitation.

If we assume exams is conducted in T days and in every days in R turn, and in every turn n class is considered for exams, then every gene is showing an exam that in day of i, i=1, 2, 3,…, T and in j turn j=1, 2, 3,…, R and in K class K=1, 2, 3,…, n is considered and every chromosome is showing an exam schedule table. For example if n=3 and R=2 and T=14 every chromosome is presented in figure 1.

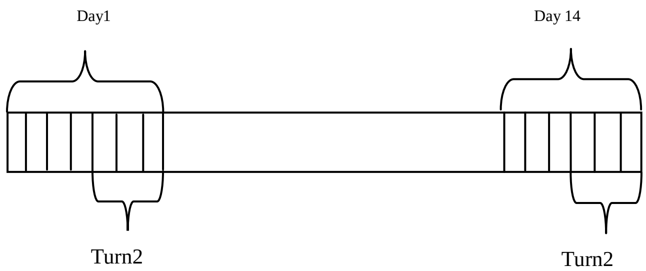

Figure 1. Genetic algorithm chromosome

Then we allocate a number to every of these Course for settling in exams schedule table, for example if 80 Course is presented in one term, we number them from 1 to 80.Also, off days because of importance of making space between the exam, are impressed on chromosome and is shown with a special number such as 100 and ultimately if a class be empty in a particular time (any exam isn't conducted in it) it is indicated with number 0. A sample of a complete chromosome is viewable in figure 2.

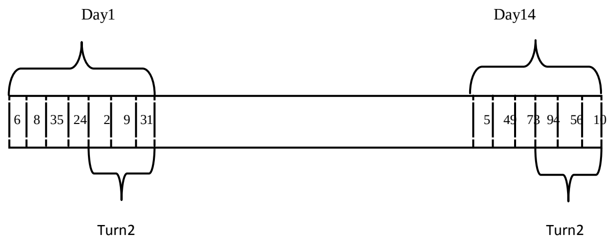

Figure 2. Exam scheduling solution representation

First generation

For first stage of the algorithm, we need to create a random population .At first we should determine off days in every chromosome. If we have an obligatory constraint (very difficult) satisfy the constraints and fixed Course related to this constraint. Then with a completely random process, put remainder Courses in rooms of this ornament. It's obvious that number of room of every ornament should be more than number of first generation and as a result the primitive population is created.

Selection and crossover

For selecting chromosomes for being pair can use of different methods, for example roulette wheel, ordinal selection, Bulletzman selection, competitive selection, etc. [17].

After inquiry all kind of methods, competitive selection method is applied for arising speed of getting better answer. There is a different method for removal, too, for instance one spotted, two spotted, multi spotted etc. which it has been used of a one spotted method in this issue. This method has its special simplicity and reduces time of implementation of algorithm by computer.

Mutation

In genetic algorithm after that a member comes to existence in new population, every gene mutates with a clear chance. Mutation of a gene means changing in that gene. Whatever chance of mutation be more, variety and distribution of population is more and this distribution, delay convergence. For this in first generations it's used of mutation chance and in last generations from smaller mutation chance [17].

Mutation methods divide to two kinds of mutation: binary mutation and real mutation, in this issue have been used of real mutation and arrangement change method for mutation. In this way that a room in chromosome ornament (one page) change its place with another room in this ornament.

Fitness function

Amount of fitness function is defined compression measure of student's exams originally in this issue and genetic algorithm is finding suitable schedule program that has least compression .For interpretation of this function, every Course that jointly persons took it , regard as two opposite electrical charge and try to reduce to least

To cite this paper: Goli A, Aghaei M. 2013. University examinations scheduling problem using genetic algorithms. *Sci. J. Mech. Ind. Eng.*, 2 (4): 82-85.
Journal homepages: http://sjmie.science-line.com/

this gravity by choosing suitable time space between them. As a result regarding to formula $F = K\frac{q_1 q_2}{d^2}$ that introduce for gravitational force two twice, we define Fitness function as below:

$$fitness = \sum_i \sum_j C_{ij} \frac{W_i W_j}{d_{ij}^2}$$

C_{ij} is number of joint bodies of Courses i and j, W_i,W_j is Course i weight and Course j weight, that determine from management and regarding to amount of Course 's importance in exams timetable (can put number of that Course 's credit). d_{ij} is time space(delay) between Course i and j exam that is determined conventionally .

$$d_{ij} \begin{cases} 1 & \text{exam of lesson i and j is conducted in one day and one turn.} \\ 6 & \text{exam of lesson i and j is conducted in one day and two turn.} \\ 100 & \text{exam of lesson i and j is conducted at more than one day space.} \end{cases}$$

Now, try to add fine in size of P_k to fitness function instead of every disrespect of limitation. So, fitness function is accounted as below ultimately:

$$fitness = \sum_i \sum_j C_{ij} \frac{W_i W_j}{d_{ij}^2} + \sum_k P_k Y_k$$

As before said ,whatever importance of constraint is more, a greater p_k fine factor is regarded for it . Binary variable Y_k, also is showing respecting or disrespecting of K th constraint.

Setting constraint in fitness function cause removal and mutation agents perform with more freedom and involve all of space of issue with more chance and take total best answer. It's clear that programs which provide all of limitation, is less fitness and as a result more suitable compared to other programs.

Computational result

In every generation choose 10 cases of best bodies (outstanding) and if it was from the best of last generations replace them, in this way in every generation 10 case of best bodies from the view of fitness, from first generation to now possess and regarding to stopping standards. After stopping algorithm, we introduce these 10 bodies as out coming of genetic algorithms.

After implementation of algorithm fitness diagram is getting as below:

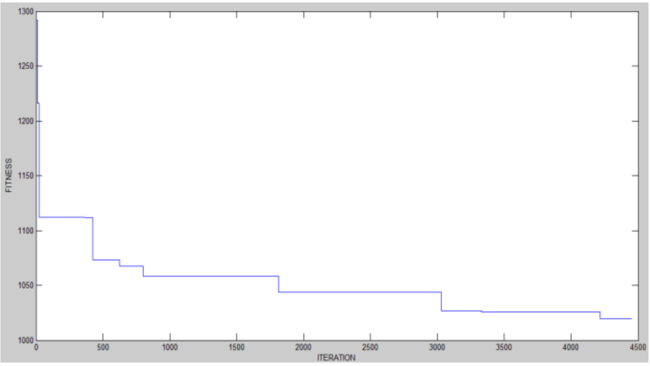

Figure 3. Best fitness of population

In order to increase convergences algorithm speed to optimal answer in every generation, import, numbers of collection "10 better" in to population. In these way outstanding features such as providing some expanding limitations and optimal answer is getting faster.

Remarkable note is that with applying such change in algorithm should increase mutation rate (chance mutation) to reduce chance of being in trap local optimum in one point.

After correction algorithm amount of fitness diagram in every generation is getting like below:

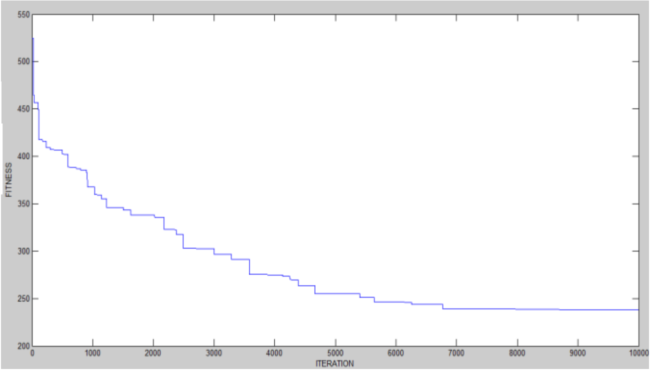

Figure 4. Best fitness of population in improved method

As you see improvement in amount of fitness function is accomplished with more speed and against first state can claim that every generation has been better than last generation and this is because of presentation of B in population of every generation.

Student satisfaction

With holding selection credit data's of all of students, for every day from exams program, find number of bodies who exams in one day and number of bodies who have 2 exams in tow nearby turn and then find the number of bodies who have two exam in two consecutive days and introduce satisfaction of students standard from exams program as below:

$$SF = 50P_1 + 5P_2 + P_3$$

That P_1 is number of all persons who exam in one time, P_2 is number of all persons who have exam in two nearby turn and P_3 is number of all persons who have two exams in two consecutive days. Exam's schedule program should set which P_1=0, unless in special position that a student is forced to have two exams in one time because of bad credit selection. put this relationship next to fitness function of algorithm help us to have tradeoff between limitation of issue and students satisfaction in some case, so if a program provides all of limitations and is suitable from fitness but it's unsuitable from student's view, break some limitations (made fitness worse) and arise satisfaction of student's (make SF less).

CONCLUSION

In this paper, we tired issue NP-COMPLETE exams schedule study and solve by one of most powerful

SJMIE
To cite this paper: Goli A, Aghaei M. 2013. University examinations scheduling problem using genetic algorithms. *Sci. J. Mech. Ind. Eng.*, 2 (4): 82-85.
Journal homepages: http://sjmie.science-line.com/

evolutionary algorithms and was shown which because of variation of issue's limitation and difference of importance amount of every in every university can design algorithm as it is responsible for all of different attitude and is not executive only for typical university. In addition by correcting genetic algorithm and participating bests of last populations in next populations, Can speed up the program improvement .comparison of exports of this algorithm with exports of other method such as has proved linear schedule, better function, facility in performance, high speed and accuracy of this algorithm.

REFERENCES

1. Abramson D., Abela J., A Parallel GeneticAlgorithm for Solving the School Timetabling Problem, Proceedings of the 15th Australian Computer Science Conference, Hobart, Australia, 1992, pp.101.
2. Beligiannisa G., Moschopoulosa C., Kaperonisa G., Likothanassisa D., Applying Evolutionary Computation To The School Timetabling Problem: The Greek Case, Journal of Computers & Operations Research, Vol. 35, 2008, pp. 1265–1280.
3. Burke E., Elliman D., Wearer R., A Genetic Algorithm based University Timetabling System, Proceedings of the 2nd East-West InternationalConference on Computer Technologies in Education, 1994, pp. 35-40.
4. Burke E.K., Elliman D.G. and Weare R.F. (1993) "A University Timetabling System Based on Graph Colouring and Constraint Manipulation", To appear in the Journal of Research on Computing in Education. Vol. 26. issue 4.
5. Davis L. and Ritter F. (1987) "Schedule Optimization with Probabilistic Search"
6. E. K. Burke, J. P. Newall and R. F.Weare "A Memetic Algorithm for University Exam Timetabling''
7. Erben W., Keppler J., A Genetic Algorithm Solving a Weekly Course-timetabling Problem, Proceedings of The First International Conference on The Practice and Theory of Automated Timetabling, Edinburgh, UK, 1995, pp. 198-211.
8. Kang L. and White G.M. (1992) "A Logic Approach to the Resolution of Constraints in Timetabling" European Journal of Operational Research 61 pp306-317.
9. Pillay N., Banzhaf W., An Informed Genetic Algorithm For The Examination TimetablingProblem, Journal of Applied Soft Computing, Vol. 10, 2010, pp.
10. Proceedings of the 3rd IEEE Conference on Artificial Intelligence Applications, February 1987, Orlando, Florida, USA, pp231-236
11. Rupert WEARE, Edmund BURKE and Dave ELLIMAN, January 1995 Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems'
12. Russell S., Norvig P., Artificial Intelligence: A Modern Approach, 3rd Ed., Prentice Hall, 2009.
13. Rossi-Doria O., Paechter B, A Memetic Algorithm for University Course Timetabling, Proceedings of the CO2004 Conference, Lancaster, UK, 2004, p 65.
14. Tripathy A. (1984) "School Timetabling - A case in Large Binary Integer Linear Programming" Discrete Applied Mathematics 35.3 pp313-323.
15. Vorac J., Vondrak I., Vlcek K., School Timetabling Using Genetic Algorithm, Technical Report, VSB-Technical University of Ostrava, Czech Republic, 2002
16. www.mshams.ir/heuristic.html
17. www.myreaders.info/html/artificial_intelligence.html
18. Yen-Zen Wang "Using genetic algorithm methods to solve course scheduling problems"
19. Tassopoulos, I. X., & Beligiannis, G. N. (2012). Solving effectively the school timetabling problem using particle swarm optimization. Expert Systems with Applications, 39(5), 6029-6040.
20. Sørensen, M., & Dahms, F. H. (2014). A Two-Stage Decomposition of High School Timetabling applied to cases in Denmark. Computers & Operations Research, 43, 36-49.
21. Dun, Y. J., Wang, Q., & Shao, Y. B. (2014). A Simulated Annealing Genetic Algorithm for Solving Timetable Problems. In Fuzzy Information & Engineering and Operations Research & Management (pp. 365-374). Springer Berlin Heidelberg.
22. Abuhamdah, A., Ayob, M., Kendall, G., & Sabar, N. R. (2014). Population based Local Search for university course timetabling problems. Applied Intelligence, 40(1), 44-53.