**JOURNAL OF CURRENT RESEARCH IN SCIENCE (ISSN 2322-5009)**
**CODEN (USA): JCRSDJ**
**2014, Vol. 2, No. 6, pp: 909-914**
**Available at www.jcrs010.com**

**ORIGINAL ARTICLE**

# A NOVEL APPROACH FOR UNIVERSITY COURSE SCHEDULING IN ISLAMIC AZAD UNIVERSITY

Fariborz Ahmadi[1], Reza Tati[2], Sara Safavi[3]
*1. Department of computer science, Yazd branch, Islamic Azad University, Yazd, Iran*
*2. Department of computer science, Miyaneh branch, Islamic Azad University, Miyaneh, Iran*
*3. Department of computer science, Yazd branch, Islamic Azad University, Yazd, Iran**
*Corresponding author email: safavis@gmail.com*

**ABSTRACT:** university course scheduling is one of the NP-hard problems that should be solved by nontraditional approach like genetic algorithm, ant colony and the same other methods. The primary goal of this problem is finding the best scheduling that satisfy teachers' time and students' time with limited number of classes. In this paper a new method based on the behavior of eurygaster is used to solve this problem. In this approach, for each solution one eurygaster based on their behavior is produced and several eurygaster constitute one partition. After creating all partitions, the main algorithm is executed to find the best solution to satisfy all of the above mentioned circumstances.int the other word, each eurygaster is a solution for the problem and the best eurygaster is absolute solution. The evaluation results of applying this approach indicate our proposed approach is faster than genetic algorithm in obtaining the best solution of the problem.

***Keywords*** **:** evolutionary computation, genetic algorithm, university scheduling.

## INTRODUCTION

Course scheduling is important job in real-life situation because of numerous universities in the entire world. Also, this problem is a NP-Hard problem that makes it hard to solve by normal algorithm because of computational complexity [1]. The main goal of this problem is finding a solution that arranges teacher, student, classroom and its time optimally [3, 4, 6]. There are several constraints to be considered in this problem. First, the number of classrooms and their capacities must be evaluated. For example, if capacity of class C is 20, the maximum number of student registering in that class should be 20. Second, teacher availability should be considered too, because teachers are not available all of the time of the week days. Third, the number of classes for each teacher and student in one day must be maximized. According to these constraints, new approach is designed by researchers based on the eurygaster life [5, 7, 8].

In the other hand, eurygaster algorithm is an algorithm developed to solve NP-Hard problems. In this algorithm, each eurygaster shows one solution and the best solution is the answer of problem. In this algorithm a set of eurygasters distribute over wheat farms and ruin them. After ruining one farm, they migrate to adjacent farm to disturb it. This routing is continued until either all of the farms are ruined or the best farm is reached. By ruining farms, we mean that farm or partition doesn't contain eurygasters that satisfy the solution of problem. Also, by migrating we mean that new partition is created to investigate whether the best solution is found or not found. The evaluation results show that our approach is faster and better than genetic algorithm in obtaining the best solution of problem.

## EURYGASTER BEHAVIORS

Eurygaster integricep is an insect pest that predominantly attacks grains, feeding on the leaves, stems and grains, reducing yield and injecting a toxin into the grains which adds a foul smell to the resulting flour, and substantially reduces the baking quality of the dough[2].

In winters eurygasters live under the plants and bushes in hillside, in several numbers and make a group. At the end of winter and at the beginning of spring when it gets warmer, these insects end their winter sleeps and get ready to move and fly to grain fields by moving over the high mountains and leaving the nests in groups. The first group by the use of its instinct finds the best and the nearest grain fields and stays there. Getting there, this group of insect sends signals

to the air to show the other groups their being there. Based on the number of eurygasters in a place, the strength of signals will be different. If the number of eurygasters in a grain field is not great, the rate of diffused signals will be little and if the number of eurygasters in a grain fields is greater, the rate of diffused signals will be increased. They diffuse these signals to show the others that reside there. So that the other groups of eurygasters understand that they should not close to the grain field which contains the first group. Of course the other groups based on diffused signals by the first group and the strength of these signals they decide if they can land and stay there or not. If the power of diffused signals is low, it means that some of the other groups of eurygasters can land and stay by the other groups which are resident there and began to eat. While the strength of the signals in the sky is high, it means that the other groups cannot land on the field(s) containing eurygasters, and they must fly to other fields in which there are no eurygasters, to live and eat.

According to the passage mentioned above, the next group of eurygasters while flying from their nests to other fields to find the best grain fields searches the best and closest ones to land and eat based on the broadcasted signals by landed group(s). This process will continue until they will find a suitable and useful grain field to eat.

We conclude that all the grain fields in a wide area will be attacked by eurygasters, because they do not gather in a one place.so, when there is not enough food in a grain field in which the eurygasters have stayed for a time, they will fly to a new field with no eurygasters according to the process mentioned above.

## EURYGASTER ALGORITHM

In this section, eurygaster algorithm is described. Solving non-linear functions are so necessary in real life today and recently researchers interested in inventing methods to solve them. The great advantage of this algorithm is that it's so easy to implement and is also inexpensive in term of memory and speed. The second advantage of this algorithm is its convergence speed compared to other methods like GA and PSO [2].

The related semi-code of the proposed algorithm is as algorithm1. This algorithm is formed by combining of 3 sub-algorithms. In each phase, we described how to apply this algorithm to university scheduling in a way it can be solved.

## INITIALIZATION

In this phase university scheduling problem is divided into three partitions and each partition is investigated separately. Also the structure of eurygaster is constituted. The structure of each eurygaster is as figure1.

| *Days* | *Time* | *Class$_1$* | *Class$_2$* | *...* | *Class$_n$* |
|---|---|---|---|---|---|
| **Saturday** | 8-10 | | | | |
| | 10-12 | | | | |
| | 14-16 | | | | |
| | 16-18 | | | | |
| | 18-20 | | | | |
| **...** | | | | | |
| **Wednesday** | 8-10 | | | | |
| | 10-12 | | | | |
| | 14-16 | | | | |
| | 16-18 | | | | |
| | 18-20 | | | | |

Figure1. a sample of each eurygaster

Each eurygaster has three-dimension structure like figure1. Each cell explain that course $C_i$ is taken by teacher $T_j$ at time $S_k$ to $E_k$ on day $D_p$. Several of these eurygaster constituted one partition. In this research search space of problem is divided into three partitions as described as follows.

- Scheduling with continuous classes just for teachers
- Scheduling with continuous classes just for students
- Scheduling with continuous classes for both students and teachers

Eurygaster algorithm and its works is decribed in algorithm1 [2].

I← the number of clusters

While I <> 0 do

1. Initialization: produce euragasters or particles according to characteristic of one partition
2. Distribution: distribute eurygasters on the regions of the partition
3. Evaluation: evaluate suitability of each eurygaster or particle depend on the problem

3.1 If the suitable result of the partition is not obtained

3.1.1. Change the position of Eurygasters in the partition

3.1.2. goto 3

3.2 If the result of the problem is not obtained

3.2.1. I--

3.2.2. goto 1

Else

3.2.3. Stop algorithm or break

End while

Algorithm1. Eurygaster Algorithm [2]

## DISTRIBUTION

In this phase, eurygasters according to their structures are produced and spread over the region of first partition. There are three probabilities in each partition. First, the answer is not found and the whole search space of partition is not investigated. In this case, some other eurygasters are produced and distributed to remaining section of partition. Second, the answer is not found and the whole search space of partition is investigated. In this case, some erygasters are created and spread over another partition. Finally, if the answer of problem is found in partition, the algorithm is terminated [2].

## SUITABILITY

In order to find the solution of the problem, eurygasters should be evaluated so that the best answer is obtained. In order to reach the best solution, we must have a tool that justifies the solution. In this paper, the following functions are used to examine the suitability of solutions.

$F_{1=} \sum_{t=1}^{n} t_{continuous}$

Where t is teacher code and if the program of teacher t be continuous, $t_{continuous}$ is 1, otherwise its value is 0. In this research, the more teacher program be continuous is better.

$F_{2=} \sum_{st=1}^{n} st_{continuous}$

Where st is student code and if the program of student st be continuous, $st_{continuous}$ is 1, otherwise its value is 0. In this research, the more student program be continuous is better.

The overall function is as follows.

$$= \begin{cases} f \\ 0, \quad if\ program\ conflicts\ for\ each\ teacher \\ 0, if\ each\ classroom\ has\ more\ than\ one\ course\ in \\ f1 + f2\ , otherwise \end{cases}$$

It is obvious that the best answer is one with highest suitability of f.

## EVALUATION RESULTS

To evaluate results, we use course scheduling of ghorveh branch of Islamic Azad University in Iran. Also, semester 93-94-1 and 92-93-2 and 92-93-1 are chosen to evaluate our approach and genetic algorithm. Figure2 and figure3 show that our algorithm has better convergence and also better speed in comparison to genetic algorithm. For better evaluation, we executed our algorithm and genetic algorithm 20 times in each semester. After that minimum time and average time to reach better solution and also better solution in each semester are recorded. Our test was runned on desktop computer with following specifications.

CPU: Xeon L3014
RAM: 8 GB DDR3
OS: windows 8

Figure2 described minimum time in each semester to reach better solution. As you can see in this figure, our algorithm has better speed than genetic algorithm. Also, figure3 shows average time to reach better solution in each semester by proposed algorithm and genetic algorithm. This figure illustrates that average time of our algorithm to obtain solution of the problem is less than genetic algorithm. Figure 4 shows the percent of reaching to best solution in 20 runs of execution. As this figure describes, our algorithm has better convergence speed in comparison to genetic algorithm. Finally, figure5 describe the best solution obtaining by both of the algorithm according to suitability function. As you can see in this figure, both algorithms reached to the same solution, namely, the best solution of problem is obtained. It is mention that conflict in teacher programs and class programs are avoided in both algorithm.

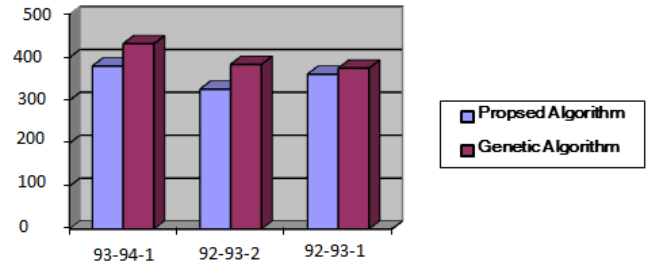


Figure2. Minimum time of running both algorithms in 20 runs in term of second

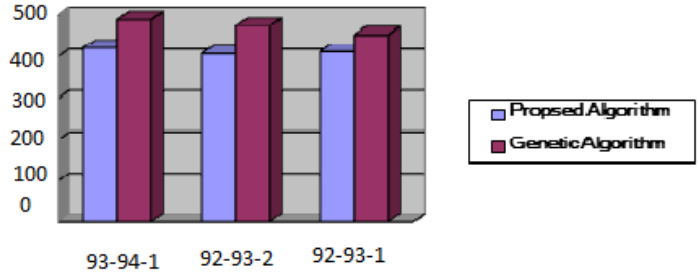


Figure3. Average time of running both algorithms in 20 runs in term of second

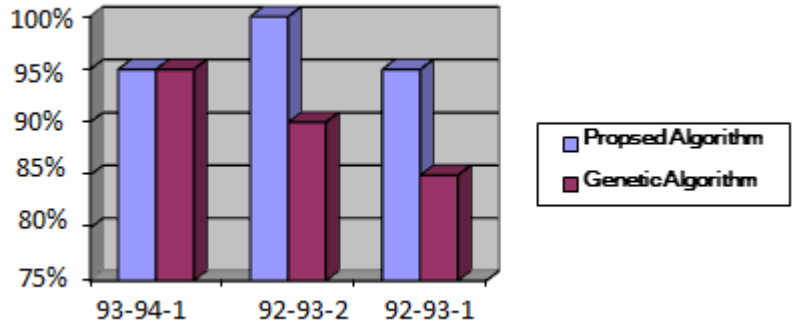


Figure4. Percent of reaching to the best solution in 20 runs

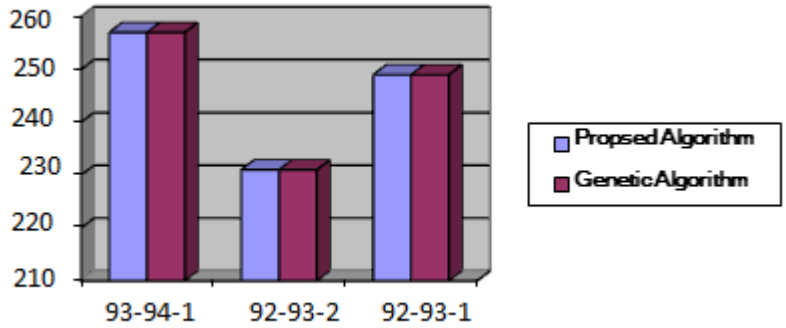


Figure5. The best value of suitability function

The evaluation results show our approach is faster and more accurate than genetic one. This approach unlike genetic algorithm lacks the local optimum so the probability of getting the more accurate solutions in this method is much more than the genetic algorithm. Moreover, in this method every space of the problem is searched for once while in the genetic algorithm every part of the problem space can be searched several times in different generations, so the rate of convergence in this algorithm is much more than the genetic algorithm. Figure 3, 4 show the convergence speed of proposed approach is faster than genetic ones. Also by figure5, it is concluded that researchers work has accurate solution in comparison to genetic algorithm.

## CONCLUSIONS

In this article, one approach based on the behaviors of eurygasters has been presented to university scheduling problem. In our approach the search space of problem is divided into three partitions and in each step of algorithm one partition is investigated. Also, there are several constraints in our proposed approach that make it difficult to solve by normal algorithm because of computational complexity. Teacher continuous program and also student continuous program are respected to obtain a solution that is appropriate for both of teachers and students. Our proposed approach uses divide and conquer method to partition the problem into several sections and then solving each section separately. This approach unlike genetic algorithm lacks the local optimum so the probability of getting the more accurate solution in this method is much more than the genetic algorithm. Moreover, our proposed method is more suitable than genetic algorithm not only in convergence speed but also in accuracy of solution. It can be proved that our proposed approach is optimal in used memory in comparison to other heuristics algorithm like genetic and PSO.

## REFERENCES

Abramson A. Constructing schoool timetables using simulated annealing: sequential and parallel algorithms. Management Science,1991: 37,98–113.

Ahmadi F. graph coloring problem by simulating eurygaster in behavior, archive of applied science research, 2014: 6,14, 175-178.س

ASc Applied Software Consultants. aSc Timetables Version 2000: 2.2000.0.76. http://www.asc.sk/rozvrhy.

Colorni A, Dorigo M, Maniezzo V. Metaheuristics for high-school timetabling. Computational Optimization and Applications Journal,1997: 16, 241-251.

Cooper T, Kingston J. "The Complexity of Timetable Construction Problems". Lecture Notes in Computer Science. 1996: 1153: 281-295.

de Werra D. An introduction to timetabling. European Journal of Operational Research, 1985:19,151–162.

Hertz A. Tabu search for large scale timetabling problems. European Journal of Operational Research,1992: 54,39–47.

Junginger W. Timetabling in germany – a survey. Interfaces,1996:16, 1666–74.