Optimization of timetabling structures based on evolutionary algorithms

Mohammad Eiman Jamnezhad, Islamic Azad university-Khoozestan Science and Technology Branch, Iran; e.jamnaezhad@gmail.com
Reza Javidan, Islamic Azad University – Beyza Branch, Iran; reza.javidan@gmail.com
Mashallah Abbasi Dezfouli ,Islamic Azad university-Khoozestan Science and Technology Branch, Iran; abbasi_masha@yahoo.com
Paper Reference Number:
Name of the Presenter: Mohammad Eiman Jamnezhad.

Abstract

Timetabling is one of the most important administrative activities that take place in academic institutions. Quite often it is done by hand or with the limited help of a simple administration system and usually involves taking the previous year's timetable and modifying it so it will work for the next year.

For the timetable, the recent growth in student numbers means that the timetable will be more constrained than ever. It is no longer good enough to use the previous year's timetable. Every year a new timetable must be produced to take account of staff, student and course changes causing a necessarily large amount of administrative work.

This problem is known to be NP-complete and as such only combinatorial optimization methods can guarantee an optimal timetable. In this paper we propose a optimized genetic algorithm for solving a university weekly courses timetabling problem. Preliminary experimental results indicate that the algorithm is promising.

Key words: Evolutionary Processing, Genetic Algorithm, Optimization, Timetabling, Crossover and Mutation operators.

1. *Introduction*

Timetabling is one of the areas that have been studied extensively in recent decade. Among different timetabling issues, course timetabling has been studied extensively due to its application. This is one of the most important and time-consuming task which takes place frequently (every year / term) in all the educational institutes. The quality of timetabling is vital on different groups such as professors, students and managers. Setting the timetabling of university is the specification of each pilot period and timing, professor and classroom on the grounds that these resources and the students' registration are taken into account [16].

Course scheduling has multi-dimensional in an educational system. The best optimization course scheduling is the one which makes use of the available educational resources. This means that regarding the available number of classes, the offered courses, the presence time of professors and lecturers, an optimized timetabling should be offered in an educational term which can meet the necessary parameters. In every educational center, there are some specific limitations and rules which constitute the entry parameters of the problem.

A timetabling capable of avoiding two courses taking place in one class at a certain time avoiding assigning two courses to one lecturer at one time and a timetabling which takes into account all the class constraints is vital in such educational system [15,7]. Course scheduling timetabling under the name of the assignment of subjects to needs, is considered with regard to the relevant resources and intended goals in time and space, as such, it meets one of the ideal goals. The specification of a timetabling is proved to be a difficult NP problem [4, 8, 3, and 15].

This means that the necessary calculation value to solve the problems increases remarkably with regard to the value of the problem. It takes a lot of time and makes the preparation of timetabling (which covers the intended goals) more difficult. Therefore, using search methods is necessary for the optimized solutions or near-optimized ones.

One of the method to solve these problems is genetic algorithm which does not operate on the function because it uses the search and also because of function value in different areas of the search space [11]. Some of the methods such as Newton Quvaci , Gradian Kanjogate , need near-optimized solutions for the start of problem-solving and this in turn in some cases requires other methods which provide near- optimized solutions.

However, on genetic algorithm and other methods such as random search the start of the problem solving from the random areas takes place in the search space [12].

Most of the works done in the field of course optimization so far have been theatrical and only in few cases they have been done practically [12]. In course scheduling technique, there are some limitations or requirements which form the main part of the problem and alteration in the requirements yield different results.

In various research, the requirements applied are few and do not show all the limitations and this does not yield the ideal results. The works done based on genetic algorithm combine different available exploratory algorithms to reach the best result in most of the cases. But none has optimized the genetic algorithm to reach the ideal result [7].

In this paper all the limitations and available requirements such as the main requirements (hard requirements) and secondary requirements (soft requirements) are studied.

In the second part of the paper the concept of course scheduling is studied and in the third part, the evolutionary processing and their applications in optimization is studied and then a method is proposed for the optimization in course scheduling. A last, the conclusion is offered.

2. Course Scheduling

The available parameters in a course scheduling system include the offered courses, the schedule for professors' presence, and the available rooms for the classes. The general procedure is as follows: A course is taught by a professor in a classrooms and this should not violate the other cases and rules which are the limitations of problem.

The limitations of a problem are divided into two types, main and secondary ones. those which are applied in the final course scheduling are the main ones and those which their application makes the course scheduling reach a better state are the secondary ones [7,15] .

Some of the main requirements have been widely used while the others have been used by only a few researchers. Some of these requirements and constraints have been considered by some researchers as main ones and by some as secondary ones [17].

For example, the capacity of classroom is considered as a main requirement by Erbben and Keppler [13], Burke and Petrovic [6], Sucha et al [19] and as a secondary one by Alvarez valdez et al [1] and Gaspero et al [14].

Erben et al and Burkeetal used the data gathered from Nottingham University but most of the researchers used unpublished data of their institutes. Both main and secondary requirements used in different scientific organizations might be private.
Daskalaki et al used an integer programming formulation in university timetabling in Greece, in this programming formulation, course scheduling is within 45 minutes. One of the studies in Patras University has considered 72 credits in 211 terms. Alvarez Valdez et al used tabu search to solve the problems of designing timetabling of the credits in Valencia University, this included 13 educational periods (terms) during each working period.
The course scheduling system of the center studied in this paper is in such a way that the number of the sessions held for each course is different and it is between 4 to 24 sessions; and it has to be held within a 12-week period and the courses which are more than 12 sessions should not be held more than two sessions a week.
In case of official vacation during this 12-week period, the makeup class should be included in the scheduling. The limitations of the course scheduling are of two types:
The main limitations consist of the followings:

- Two different courses should not be held at the same time and in one place (class).
- Two different courses offered by a professor should not be held at the same time (place is not considered here).
- The number of courses offered (the number of professors) in the same time period should not be more than the total number of places (classrooms).
- The timetabling for the professors' presence should not be out of the time period specified by the professors.
- The courses which are to be offered by educational chart during one term should not be held at one time (should not meet at the same time)
- Doubling a course should be done by doubling the time.

And the secondary type includes the followings:

- The hours of a course in a term should be after each other.
- The professors' timetabling should be without interruption.
- Holding the make-up classes for the days off and official holidays.
- The free interval between students timetabling should be the minimum time.
- The classes timetabled for a period should be in the same room or same time during the day timetabled.
- The courses which are held less than 12 sessions a term should be put at the end of the timetabling plan.

3. Algorithm and its methodology

3. 1. Evolutionary Algorithm

In the evolutionary algorithm set, genetic algorithms are multi-purpose and powerful optimization tools which model the principles of the evolution [10].
They are capable of offering ideal solutions even in the most complex research atmospheres. They are implemented on a set of encoded solutions, these solutions are chosen based on quality and then it is used as a basis to provide new solutions (this is sometimes done by changing the available components). In the past, the research mechanism was not related to the kind and range of the components. In the other word, the combination and the alteration of the components were done without knowing what the appropriate solutions were. But using independent operators have been documented to yield good results [5, 7]. Genetic algorithm uses 3 major rules to make the next generation out of the present population in each phase.

Selection rules: select some individuals as parents who are involved in the next generation population.
Crossover rules: combine the parents for the creation of the next generation.
Mutation rules: uses the random alterations for the individuals' parents to give birth to the children.

3. 2. The proposed method

The different parts of genetic Algorithm are explained in timetabling issues.

3. 2. 1. Gene and Chromosome

In this problem, each chromosome is a unique solution for the problem. It changes due to the other parts of the algorithm to reach its best. Regarding the main and secondary limitations, the fields forming gene include the course code, professor code, term offered, the number of hours of each course, the number of places (classrooms). They are represented as follows: Professor Code, the character offered, the term offered.

The offered character means that the course X is offered by professor Y. putting the character offered generates the course details after getting the final results. Character can be called the professor virtual code.

Chromosome consists of this kind of gene. The arrangement pattern of chromosome consists of a multi- dimensional arrangement of objects: the first dimension consists of 12 parts which means the 12-week educational (course) period. The other dimension is the 6 days of the week: each space (day) of this dimension includes in classes which are divided into 6 time intervals. Each course offered by professor (gene) is placed in one of the empty spaces. The good point about this is the possibility of holding a class in a classroom which is assigned by the person in charge and regarding the capacity of the classroom. Using virtual codes shortens the length of the genes and as a result decreases the additional computations which change into a range of numbers:

3. 2. 2. Selection and Crossover

The selection operator imitates the performance of the strongest animal survival in the nature; it means that it assigns more probability to the stronger animal and less probability to the weaker one for their survival.

To do the crossover N percent of the total population which are the couples is selected so that by using these parents, the next generation can be reproduced.

Transposition in this problem takes place when chromosome pair is only reproduced as the producer of one offspring. Presumably, the other unit in chromosome is the professor; therefore, a chromosome is a combination of professor chromosomes. The probability of the presence of each chromosome from the selection phase to take part in the next generation is 50% percent. Each chromosome is selected randomly and is transferred to the offspring chromosome so that all the tiny parts of the parents' chromosome are transferred to the offspring.

The N percent of the middle generation population is obtained by repeating the same method and (100-n) percent of the previous generation chromosomes with the high value is transferred to the middle generation population.

3. 2. 3. Mutation

Mutation plays a secondary important role in the performance of genetic algorithm. In smart genetic systems, the performance of the mutation prevents trapping into local mutation as a random movement in the search space. In timetabling problem, the way mutation acts is very important: using the improper mutation operator destroys the gene and also chromosome. It means that it causes the limitation of the problem to be affected and generally to change. Therefore, using a proper method can make this job easier and facilitate the problem development.

In the present problem, mutation is considered with different values which are discussed in details later. The M percent of the total chromosomes which are put next to each other like a thread is chosen and rescheduled.

3. 2. 4. Improvement in genetic algorithm

As mentioned before, genetic algorithm consists of several parts. The two most important part of the genetic algorithm is mutation and crossover: the proper implementation of these two parts yields ideal results. Besides, mutation and crossover rate are very important.

Before considering and implementing the algorithm the rate of mutation and crossover should be considered. First, the rate of crossover was considered seventy percent and regarding the obtained optimization algorithm the crossover rate of sixty and eighty percent were tested in this paper. The mutation value should be as such that few percentages of the genes are affected [12]. By computing the total number of genes the following value can be obtained.

The value of chromosome = the number of classrooms * 6 * 7 * 12

The total number of genes = the number of generations * the value of chromosome

If the number of classes is taken as the minimum number that is 10 and the number of population is taken 500 on the average, the following is obtained:

The total number of genes =2520000

The rate of mutation	The number of genes affected
10%	252000
1%	25200
0.1%	2520
0.01%	252
0.001%	25.2
0.0001%	2.52

Table 1: the value of mutation

Optimizing Genetic Algorithm is done by a method named substitution method. In this method, when the algorithm growth stops, no increase is found in the value of the best available chromosome; few new chromosomes are produced and replace the few chromosomes which are selected randomly. The reason why this happens is that after producing different generations, the arrangements of genes in different chromosomes desire a proper pattern, since the new generation is produced through the previous generation chromosomes, and causes a lot of similarity between different chromosomes.

The result can be better displayed by one example and assuming each professor (lecturer) assigns 6 hours for presence and the course (education) system assigns four courses for the lecturer and each course is included in one hour, there are 600 different states in which 72 optimized states exist. They (72 optimized states) are produced after producing different generation and after considering soft and hard requirements. There is the possibility that most of these states are not put into use or put into chromosomes which do not have higher values.

The presence of new chromosome in the present generation causes new states and the chromosomes value is increased. As it is clear from figure 1-a, the maximum value of chromosome is constant from the 150 generation up to the 500 generation, while replaced chromosomes are present, the value of chromosomes increases again. In figure 1-b which is related to the total mean of the population in each generation, the mean of value decreases at first and then it increases.

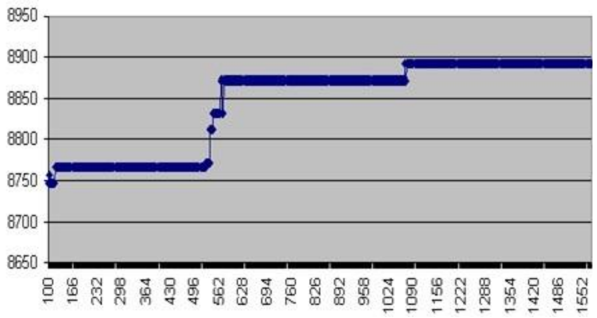

Fig. 1-a

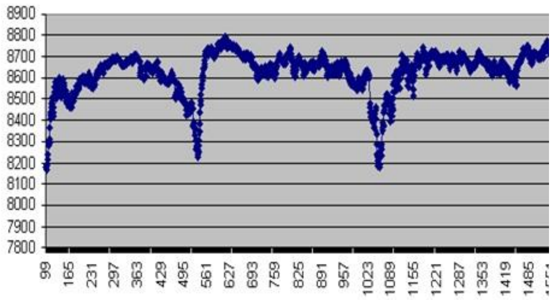

Fig. 1-b

The implementation of the timetabling is done with the populations of 100, 200… 1000 and all the results are obtained using these implementations. For example, the maximum value difference in several generations is as follows:

Generation	800	900	1000
Optimization algorithm - Main algorithm	1.21%	3.31%	2.57%

Table 2, maximum value difference

3. 2. 5. Considering the rate of mutation and crossover

Some experiments were done to determine the rate of mutation and crossover using optimization algorithm, they determined the rate of crossover to be 70, 60 and 80 percent and the value of mutation to be 0.1, 0.01, and 0.001 percent. These analyses were done on different generation and gave the result: operating on low population yielded imperfect results and in most of the cases made the degree of the algorithm growth negative.

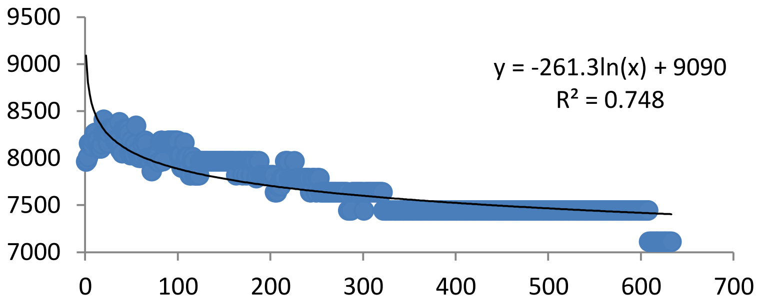

Fig 2

Table 3 shows that the growth of algorithm value is in direct proportion to the number of population. Also selecting the proper rate of mutation and crossover to reach an ideal degree is very critical.

	main	max	Generation
0.6&0.00001	85.01%	90.84%	734
0.6&0.0001	85.21%	89.43%	1039
0.6&0.001	82.71%	87.97%	1000
0.7&0.00001	85.63%	94.07%	1103
0.7&0.0001	83.13%	93.76%	1936
0.7&0.001	82.45%	88.76%	1407
0.8&0.00001	84.43%	91.72%	1533
0.8&0.0001	84.02%	90.37%	1197

0.8&0.001	84.22%	89.48%	1808

Table 3

The type of the appropriate selection function is very important in genetic algorithm and can have a great affect on the optimization. Some types of different selection functions were considered in optimization algorithm which yielded the results in table 4. As it is clear in Table 4, roulette wheel algorithm is a good method to reach optimization, although the competitive algorithm yields the same result.

CR = 0.7 ,MR = 0.0001	Equal opportunity	Roulette Wheel	Grading	Tournament
Min Value	83.96%	85.21%	87.30%	85.68%
Max Value	93.86%	***96.20%***	91.20%	94.95%

CR = 0.7 ,MR = 0.00001	Equal opportunity	Roulette Wheel	Grading	Tournament
Min Value	86.36%	85.27%	85.27%	86.41%
Max Value	95.47%	95.11%	90.68%	***95.58%***

Table 4

4. Conclusion

In this paper, due to the complexity of course scheduling problem and the presence of different requirements and limitations, an optimized algorithm was implemented and considered in the form of a genetic algorithm. This was done in a way that the maximum number of requirement such as main and secondary ones were applied in algorithms to reach a higher degree of optimization . Unfortunately the practicality of course scheduling alone can be a good reason for its applicability in an optimum level. It is also noteworthy that the practicality of a timetabling depends on views and institutes standards.

The main idea of genetic algorithm is taken from nature: therefore there are lots of ideas in the nature which can be applied to develop the genetic algorithm. Optimization algorithm offered is one example. Immigrants who enter the environments under study from other environments influence the environments and improve them. The analyses done on algorithm indicate that the number of population in the environment has a direct relationship with the maximum value of chromosome.

If the rate of mutation and crossover is selected inappropriately the growth of algorithm acts the reverse and negatively at the end. In figure 3 the optimization algorithm is shown with the population of 500 and the rate of crossover of 80 percent and the value of mutation of %0.1. Table 4 shows that the combination of different mutation and crossover rate with different selection algorithm can yield different and optimum results.

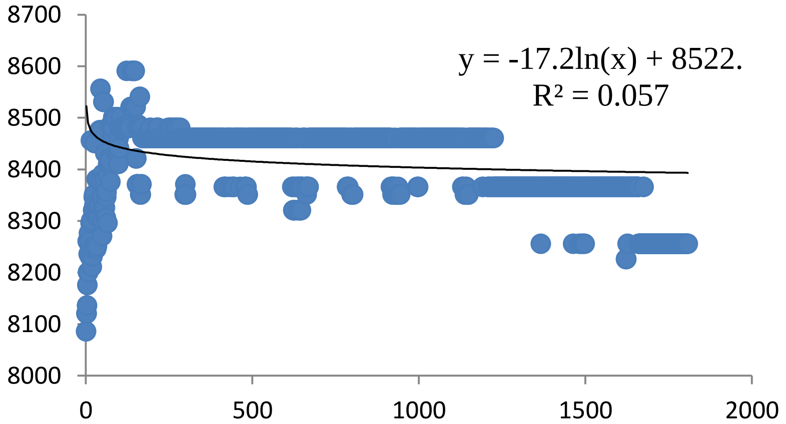

Fig 3 Crossover rate: 80%, mutation rate: 0.01%, population: 500

References

[1] Alvarez-Valdes, R., Crespo, E., Tamarit, J.M., 2002, *Design and implementation of a course scheduling system using tabu search,* European Journal of Operational Research 137 (3), 512–523.

[2] Bruns R., 1993, *Knowledge-Augmented Genetic Algorithm for Production Scheduling,* IJCAI '93 Workshop on Knowledge based Production Planning, Scheduling and Control.

[3] Burke E. K. and Kendall G. and Soubega E.., 2003, *" A tubu search hyperheuristic for timetabling and rostering,* Jornal of Heuristics, vol. 9, no.6.

[4] Burke E. K. and Newall J., 1999, *A multistage evolutionary algorithm for the timetabling problem,* IEEE Trans. Evol. Comput. Vol. 3, no. 1, PP. 63.

[5] Burke E.K. and Elliman D.G. and Weare R.F., 1994, *A Genetic Algorithm for University Timetabling,* AISB Workshop on Evolutionary Computing, Leeds.

[6] Burke, E.K. and Petrovic, S., 2002, *Recent research directions in automated timetabling,* European Journal of Operational Research 140 (2), 266–280.

[7] Causmaecker P. and Demeester P. and Vanden Berghe G., 2009, *A decomposed metaheuristic approach for a real-work university timetabling problem,* European Jornal of Operational research, Vol 195,pp. 307-318.

[8] Cooper B. and Kingston H., 2006, *The Complexity of Timetabling construction Problem,* Basser Department of Computer Science The University of Sydney Australia.

[9] Daskalaki, S., Birbas, T., Housos, E., 2004, *An integer programming formulation for a case study in university timetabling,* European Journal of Operational Research 153(1), 117–135.

[10] Davis L., 1991, *Handbook of Genetic Algorithms,* Van Nostrand Reinhold.

[11] Drexl, A., Salewski, F., 1997, *Distribution requirements and compactness constraints in school timetabling,* European Journal of Operational Research 102 (1), 193–214.

[12] Enzhe Yu and Ki-Seok Sung., 2002, *A genetic algorithm for a university weekly courses timetabling problem,* Intenational transactions in operational research, Intl. trans. In Op. res. 9 2002 703-717.

[13] Erben, W., Keppler, J., 1996, *A genetic algorithm solving a weekly course-timetabling problem,* Practice and Theory of Automated Timetabling Lecture Notes in Computer Science, 1996, Volume 1153/1996, 198-211,

[14] Gaspero, L.D., Schaerf, A., 2001, *A case-study for Easy-Local++: The course timetabling problem,* Research Report, Universita degli Studi di Udine, Italy.

[15] Grigorios N. Beligiannis and Charalampos B. Moschopoulos and Georgios P. Kaperonis and Spiridon D.Likothanassis., 2008, *Applying evolutionary computation to the school timetabling problem: The Greek case,* Computer & Operation Research, Vol. 35, pp. 1265-1280.

[16] Mohamed Tahar Ben Othman, Senior Member IEEE, 2010, *Ant Universal Tool for University Course Schedule Using Genetic Algorithm,* (IJCNS) International Journal of Computer and Network Security, Vol. 2, No. 6.

[17] Pongcharoen P. and Promtet W. and Hicks C., 2008, *Stochastic Optimisation Timetabeling Tool for university course scheduling,* int. j. production Economics, Vol 112,pp.903-918.

[18] Rossi-Doria O. and Blum C. and Knowles J. and Samples M.and Socha K. and paechet B., 2002, *A local search for the timetabling problem,* Proceeding of the 4th international conference on the Practice and Theory of Automated Timetabling, pp. 124-127 Gent , Belgium.

[19] Socha, K., Sampels, M., Manfrin, M., 2003, *Ant algorithms for the university course timetabling problem with regard to the state-of-the-art,* Proceedings of the Third

European Workshop on Evolutionary Computation in Combinatorial Optimization. Springer, UK.