A NOVEL GENETIC ALGORITHM TECHNIQUE FOR SOLVING UNIVERSITY COURSE TIMETABLING PROBLEMS

Othman MK. Alsmadi [a], Za'er S. Abo-Hammour [b], Dia I. Abu-Al-Nadi [c], Alia Algsoon [d]

[a,c]Department of Electrical Engineering, University of Jordan, Amman, Jordan
[b,d]Department of Mechatronics Engineering, University of Jordan, Amman, Jordan

ABSTRACT

In this paper, as a machine learning or system modeling, a novel genetic algorithm (GA) approach for solving university course timetabling problem is presented. The designed timetabling is free of any hard constraint violations and satisfies most of the soft constraints as much as realistically possible. When compared with other methods, the followings were the advantages: 1) the number of soft constraint violations is less, 2) the use of external rooms is eliminated, and 3) the number of instructors' overload is less.

1. INTRODUCTION

Lecture (course) timetabling problem (TTP) is the problem of arranging a sequence of meetings between teachers and students in predetermined time slots satisfying a set of constraints of various types. This involves lectures, courses, students, time periods and rooms. Classes (lectures) are predetermined every semester by different departments based on existing resources, which include the instructor workable load and the student course need [1, 2].

The TTP includes various types of constraints where most of them fall into either of the following categories:

1- Hard constraint (HC): This where constraints can't be violated where some maybe listed as [3]:
- Instructors can't lecture simultaneous lessons.
- Different classes can't be held in the same room at the same time.

2- Soft constraint (SC): This is the type where preferences are accounted for where some maybe listed as [3]:
- Instructors' preferred time slots.
- Instructors' preferred rooms.
- Same level courses taking different timeslots.
- Courses and prerequisites taking at the same time.
- Instructors' overload credit hours.

Several heuristic tools evolved in the past decades to facilitate solving general optimization problems that were previously difficult or impossible to solve [1,2,4,5]. These tools include evolutionary computation, simulated annealing, tabu search, particle swarm, case based reasoning (CBR), ant colony optimization, genetic algorithms and so forth. Each method simulates physical, biological and natural phenomenon. However, timetabling solutions obtained using such methods were either incomplete, in terms of considering all important hard constraints, or they work for only some specific problems [1,2,4,5,6,7,8,9]. On the contrary, the work presented in this paper provides the most comprehensive solution for students, teachers, and the university.

2. GENETIC ALGORITHM METHODOLOGY

Genetic algorithms are heuristics or approximate algorithms that are based on natural selection and evolution [10]. The GA is performed as following [11]:

1- Initial population:

We generate several individuals that are free from hard constraints' violation. Each individual consists of a number of entries (n_p) where each entry includes a number of parameters that are required to form a suitable timetable. In our approach, each entry includes the following parameters: subject, section, instructor, time, and room. For each course entry, where (S_i) represents subject, (S_{ei}) represents course section, (D_i) represents instructor (or doctor) for the course, (T_i) represents time, and (R_i) represent the time and room of the lecture. Hence, part of an individual with three entries can be illustrated as seen in Figure 1.

S_1	S_{e1}	D_5	T_{11}	R_1	S_1	S_{e2}	D_2	T_4	R_{10}
S_3	S_{e1}	D_5	T_1	R_3					

Figure 1. Individual representation with $n_p = 3$.

In our example, all of the data is retrieved from a data base where subjects, instructors, times, and rooms are all given in real numbers from 1 to N_i (N_i represents the maximum number). Thus, the individual in Figure 1 is represented in our method as shown in Figure 2.

1	1	5	11	1	1	2	2	4	10	3	1	5	1	3

Figure 2. Actual individual representation.

978-1-4577-0690-5/11/$26.00 ©2011 IEEE

As it can be seen, for each course in the given individual, we need a suitable number of sections. The number of these sections depends on the number of students enrolled in the same course level (taking into consideration the failed students from the next level). As a result, the number of sections-per-subject-per-semester is estimated based on the following formula:

$$n_{sss} = \left(\frac{n_l + (f_r . n_{l+1})}{50} \right) / 2 \quad (1)$$

where n_l is number of students in the course level, n_{l+1} is number of student in the next level, f_r is failure rate for this course, 50 is the assumed room maximum seat capacity, and 2 is the number of semesters per year.

The length of each individual, which is a solution (required data), in the initial population is given by

$$L_{IP} = \sum_{i=1}^{n_p} n_c(i) * n_{se}(i) \quad (2)$$

where n_c represents the number of courses and n_{se} represents the number of sections for each course. All of the individuals in the initial population are free of any hard constraint violations. On the other hand, the hard constraints that are mainly considered are listed as:

- No instructor can give two or more lectures simultaneously.
- No two or more lectures can be held in the same room at the same time slot.
- The maximum number of time periods per week is15, which may not be exceeded.
- Each course must be assigned to an instructor in a specific group that usually teaches the course.
- Classes of type theory need 3 credit hours separated during the week but classes of type "lab practice" need 3 continuous hours in one day.
- Classes of type "lab practice" are assigned to a certain laboratory, not in lectures room.

2- Fitness evaluation:

As an objective function, fitness evaluation is used. The fitness is calculated for each individual as following:

$$F = \sum_{i=1}^{n_p} w_h \left(1 - \frac{n_{hv}(i)}{n_{hv\max}}\right) + w_s \left(1 - \frac{n_{sv}(i)}{n_{sv\max}}\right) \quad (3)$$

where w_h is the weight of the HC, n_{hv} is the number of HC violations, $n_{hv\max}$ is the max. possible number of the HC violations, w_s is the weight of the SC, n_{sv} is the number of the SC violations, $n_{sv\max}$ is the number possible SC violations.

The soft constraints that will be used in the proposed timetabling approach are listed as follows:

- *Time soft constraints* (TSC)
  Teachers may prefer specific times for their lectures.
- *Room soft constraints* (RSC)
  The room soft constraint allows teachers to choose specific rooms for their lectures.
- *Level soft constraint* (LSC)
  Same level courses maybe given at the same time or at different timeslots.
- *Prerequisites soft constraint* (PSC)
  The courses and their prerequisites are scheduled so that students can make their registration easily.
- *Instructor overload soft constraint* (IOLC)
  This constraint is used to obtain a suitable course distribution for all instructors to avoid overloads.

As a result, Equation 2 can be rewritten such that all of the constraints are included:

$$\begin{aligned} F = {} & w_{ht}\left(1 - \frac{n_{htv}}{n_{htv\max}}\right) + w_{hr}\left(1 - \frac{n_{hrv}}{n_{hrv\max}}\right) + w_{ts}\left(1 - \frac{n_{tsv}}{n_{tsv\max}}\right) \\ & + w_{rs}\left(1 - \frac{n_{rsv}}{n_{rsv\max}}\right) + w_{pre}\left(1 - \frac{n_{pre}}{n_{pre\max}}\right) \\ & + w_{ls}\left(1 - \frac{n_{level}}{n_{level\max}}\right) + w_{load}\left(1 - \frac{n_{load}}{n_{load\max}}\right) \end{aligned} \quad (4)$$

where w_{ht} is the HC weight, n_{htv} is the number of violation time HC, $n_{htv\max}$ is the max. possible number of time HC, w_{hr} is the room HC weight, n_{hrv} is the number of room HC violation, $n_{hrv\max}$ is the max. number of possible violations in room HC, w_{ts} is the room SC weight, n_{rsv} is the number of violation in room SC, $n_{rsv\max}$ is the max. number of room SC violations, w_{pre} is the subject weight and its prerequisites taking same time, n_{pre} is the number of prerequisites SC violations, $n_{pre\max}$ is the max. number of violations in number of sections per subject, w_{ls} is the subject level SC weight, n_{level} is the number of violation in subject level SC, $n_{level\max}$ is the max. possible number of violations in number of section per subject, w_{load} is the instructor load SC weight, n_{load} is the number of violations in this SC, $n_{load\max}$ is the max. possible number of violations in the instructor's load.

3- Selection:

In the selection process, we use the rank based selection type where a prescribed number of parents' individuals are chosen with highest fitness according to the rank based ratio, R_{rb}. Then, the mating process is performed by choosing parents at random from this subpopulation, which is of size $R_{rb}*N_p$ where N_p is the number of individuals in the initial population [11]. The highest 10% of the initial population individuals will be selected to enter the mating pool.

4- Crossover:

In our research, we randomly select the individuals from the mating pool and then we define a random number. If the random number value is less than a certain crossover probability, then we do crossover, otherwise they pass without any change. This is illustrated as in Figure 3.

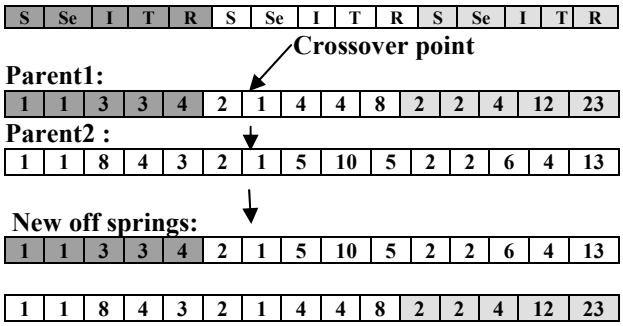

Figure 3. Single point crossover.

5- Mutation:

The new offspring, after crossover, is randomly chosen to perform the mutation. This process is governed by a certain probability, *Pm*. That is, if we choose the second offspring after crossover seen in Figure 3, then Figure 4 shows how it will be after mutation.

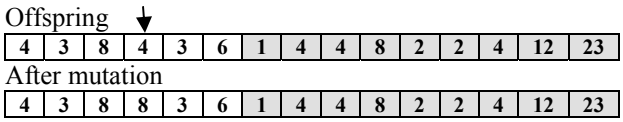

Figure 4. Mutation process.

6- Replacements:

The general replacement scheme is used in our approach where the number of elite (best) parents passed (N_e) depends on the replacement factor (R_f) and number of population individuals (N_{pop}) such that :

$$N_e = (1 - R_f) N_{pop} \quad (5)$$

On the other hand, the number of the elite offspring passed (N_o) depends on the replacement factor yielding $N_o = R_f * N_{pop}$. Hence, the total population size (P_{total}) is given as N_o plus N_e. As we can see, in each generation, we replace the individual by the new solution keeping the same number of individuals in each generation (using a replacement factor 0.1). In addition to that, we make sure that the best parents are still present in the next generation along with the best children. This is performed in order to guarantee converging into the best solution and not diverging into other solutions.

7- Termination:

The GA is terminated when some convergence criterion is met. In the proposed timetabling approach, the GA will be terminated when the fitness becomes equal to or greater than 0.99 or when the maximum number of generations is reached.

3. SIMULATIONS AND RESULTS

The proposed GA has been applied to the faculty of engineering and technology at the University of Jordan (JU). Table 1 presents the general information used by the GA.

Table 1. General data used by the GA.

No.	Operator	Quantity/ Type
1	Number of individuals	1200
2	Crossover probability	0.7
3	Mutation probability	0.1
4	Number of generations	500
5	Selection mechanism	Rank based selection
6	Crossover type	Single point crossover

It is important to mention that the rates of the crossover and mutation were selected based on the result suitability obtained by the different simulations.

The proposed GA timetabling approach was tested for the faculty of engineering and technology at the University of Jordan for the 2nd semester, 2008-2009. Its performance was compared with the already used hand prepared timetable. Table 2 presents the comparison results where it can be seen that the GA method provides better results.

Table 2. Hand and GA prepared results comparison.

Type of violations	Hand Prepared	GA Prepared
Time hard violations	0	0
Room hard violations	0	0
Instructors' overload	100	4
Same level course/same time	180	120
Prerequisites violations	256	250
Sections outside faculties	7	0

To better investigate the new approach, we have tested each constraint with different weights (w_s presented in Equation 2) to observe the number of violations in each constraint. Knowing that the hard constraints must be satisfied (as seen in Table 2), Table 3 shows the number of violations (N_v) in the time soft constraint against different weights. It is also seen that the number of violations in each constraint (performed at different runs) is almost the same against the change in its weight. For example, the maximum number of violations in time soft constraint (TSC), which is 497, at weight of 0.7 is not very much different from the minimum number of violations, which is

Table 3. Number of violations (N_v) against weight (w_s).

w_s	TSC	RSC	LSC	IOLC	PSC
	N_v	N_v	N_v	N_v	N_v
0	480	536	354	5	333
0.1	474	558	355	5	332
0.2	484	554	354	5	336
0.3	493	560	346	5	330
0.4	489	550	350	5	334
0.5	490	546	351	5	340
0.6	486	552	343	5	331
0.7	497	542	340	5	331
0.8	489	554	343	5	334
0.9	481	558	347	5	336
1	482	550	342	5	333

474, at weight of 0.1. In fact, this is an indication that the sensitivity (S_v) of our approach to weight changes is very low. Hence, the great advantage is that there is no need for a user experience. To check the exact S_v, let the sensitivity be given by the following equation

$$S_v = \frac{nv_{\max} - nv_{\min}}{nv_{\max}} * 100\% \quad (6)$$

where $nv_{\max}$ is the maximum number of constraint violations and $nv_{\min}$ is the minimum number of constraint violations. S_v for the time soft constraint is obtained as $(497 - 474/497) * 100\% = 4.63\%$. It is important to mention that the sensitivity for the hard constraints (time and room) is 0% for any hard constraint's weight. As a matter of fact, the proposed approach is performed with the following advantages (in comparison with the other recently published work):

1- None of the individuals in the initial population includes any hard constraint violation, where on the contrary, Al-Milli [7] have no limitation on the instructor-time hard constraint.
2- Room availability is considered as one of the hard constraints so that all of the courses will be offered in the rooms of the specified department, on the contrary to the work proposed by Dammak et al. [3] and Aladag et al. [8].
3- In the process of preparing the schedule, no need for function modification to correct the timetable as performed in the work of Burke and Newall [9].

4. CONCLUSION

A TTP GA solution is presented in this paper. The proposed GA achieves a successful role in designing an optimal timetabling that meets all proper specific requirements. The designed timetabling satisfies all of the hard constraints and most of the soft constraints as much as realistically possible. Comparison of the proposed approach to other work shows that the new method obtains the most comprehensive solution. A great advantage of the proposed approach is that there is no need for a user experience. It is important to mention that this work has been compared to more of recently published work, as stated in the introduction. However, each problem was considered in a different manor, due to different resources and different parameters.

REFERENCES

[1] S. Yang, S. Jat, Genetic Algorithms with Guided and Local Search Strategies for University Course Timetabling. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 41 No. 1, pp. 93-106, 2011.

[2] A. Chaudhuri, K. De, Fuzzy Genetic Heuristic for University Course Timetable Problem. Int. J. Advance. Soft Comput. Appl., Vol. 2, No. 1, pp. 100-123, 2010.

[3] A. Dammark, A. Elloumi, H. Kamoun, Course Timetabling at Tunisian university. J. Syst. Scisyst Eng. Vol. 17, No. 3, pp. 334-352, 2008.

[4] S. Sun, F. Zhuge, J. Rosenberg, R. Steiner, G.D. Rubin, S. Napel, Learning-enhanced simulated annealing: Method, Evaluation, and application to lung nodule registration, Applied Intelligence, Vol. 28, pp. 83–99, 2007.

[5] M.K. Tiwari, A. Mukherjee, R. Shankar, Application of a kin selection based simulated annealing algorithm to solve a complex scheduling problem. International journal of computer integrated manufacturing, Vol. 18, No. 8, pp. 671 – 685, 2005.

[6] D. Chen, P. Burrell, Case-Based Reasoning System and Artificial Neural Networks: A Review, Neural Computing and Applications, Vol. 10, pp. 264–276, 2010.

[7] N.R. AL-Milli, Hybrid Genetic Algorithms with great deluge for course timetabling. IJCSNS International Journal of computer science and network security, Vol. 10, No. 4, pp. 283-287, 2010.

[8] C.H. Aladag, G. Hocaolglu, M.A. Basaran. The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. Expert systems with applications, Vol. 36, pp. 12349-12356, 2009.

[9] E. Burke, J.P. Newall, Solving examination timetabling problems through adaption of heuristic ordering. Annals of operations research, Vol. 129, pp. 107-134, 2004.

[10] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 1989.

[11] Z. S. Abo Hammour, A Novel Continuous Genetic Algorithms for the Solution of the Cartesian Path Generation Problem of Robot Manipulators. In Robot Manipulators: New Research. Nova Science Publishers, Inc. New York, 2005.