Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 10, No. 2, May 2019), Pages: 109-117
www.jacr.iausari.ac.ir

Using Artificial Fish Swarm Algorithm to Solve University Exam Timetabling Problem

Mehdi Yadollahi✉**, Seyedeh Sedigheh Razavi**
Department of Computer Engineering, Ayatollah Amoli Branch, Islamic Azad University, Amol, Iran
m.yadollahi@iauamol.ac.ir; mahsa_razavi88@yahoo.com

Manuscript ID: JACR-1902-1662 **Received: 2019/02/03; Accepted: 2019/11/08**

Abstract

The timetabling problem consists in scheduling a sequence of courses in times period between teachers and students, satisfying a set of constraints of various types. The problem of University Exam Timetabling problem is one of the complex combined problems that universities worldwide struggle with for several times over the course of the year. The objective in this problem is to assign the student exams to times period and rooms in order to meet a series of constraints. The university exam timetabling problem is of difficult problems due to the very large search space, and innovative algorithms are more used to solve it. Several methods have been suggested to solve this problem so far. For the first time, the aim is to present a new method in the present study for university exams timetabling using an artificial fish-swarm algorithm (AFSA), then compare this method with other algorithms in order to evaluate the efficiency of the proposed method. The simulation results indicated that the proposed method is of high efficiency.

Keywords: *University Timetable, Exam, Artificial Fish-Swarm Algorithm*

1. Introduction

The university exam timetabling problem is one of the complex combined problems that universities around the world struggle with for several times over the course of the year[1]. The goal in this problem is to assign the student exams to a number of timeframes and rooms in order to meet a series of constraints. These constraints can be generally divided into two categories of hard and soft constraints. Hard constraints are the ones that must be certainly observed and lack of compliance with these constraints leads to invalid solutions. Therefore, hard constraints should not be violated in any way. However, soft constraints are the ones that, if satisfied, enhance the quality of the solution and incompliance with them does not invalidate the responses [2].

Although this definition accounts for wide range of problems (such as the preparation of timetable for school and university, timetable of staff working hours for various departments, schedule of airplane flight or train departure, a sports event schedule, etc.), most articles have focused on the timetable problem for schedule design for classes or exams in educational environments[3].

The university exams timetabling problem is referred to the process of assigning university exams to time periods and rooms in order for meeting most of the existing limitations[4]. The most important issue in the timetable problem is the constraints that are divided into two groups: hard constraints and soft constraints.

Hard constraints include:

- Exam interference: At a given time period, each student can have a maximum of one exam.
- Suitable room: Each exam must be held in an appropriate room.
- Room interference: At a given time period, a maximum of one exam can be held in each room.

Soft constraints include:

- Students will have a maximum of one exam in a day.
- The interval between the exams of each student should be at least one day.

Designing appropriate timetables for exams that can be reasonably applicable (meeting hard constraints), while satisfying the professors and students (meeting soft constraints) will be very difficult and time consuming. In practice, the constraints imposed by various universities are different. These differences are more common prominent in soft constraints, and may even exist at different faculties of a university[5]. These differences make it more difficult to create a common software or even a specific algorithm to design a suitable timetable. A software package for a problem may create appropriate responses, but it may not solve the other problem well[6].

The formulation of the university exam timetabling problem is presented in table 1. In this table, the values of R, t, E, S, and W are given as inputs and the aim is to obtain the values of matrix X.

Table 1. Formulation of the university exam timetabling problem

Description	Formulation
A set of rooms	$R = \{r_1, r_2, ..., r_m\}$
A set of time periods (10 days and 3 time periods per day)	$T = \{t_1, t_2, ..., t_{30}\}$
A set of exams	$E = \{e_1, e_2, ..., e_n\}$
A set of students	$S = \{s_1, s_2, ..., s_p\}$
Types of rooms (class, laboratory)	$W = \{w_1, w_2\}$
A matrix showing that the exam x_{ij} is held in room i and time period j	$X_{m*30} = \{x_{ij}\}$
Each exam requires a room with a special feature to hold	$for\ each\ e_i \in E \rightarrow w_j \in W$
Each room has a special feature	$for\ each\ r_i \in R \rightarrow w_j \in W$
Each student must participate in a set of exams	$for\ each\ s_i \in S \rightarrow E' \subseteq E$

So far, several methods have been proposed to solve the timetabling problem. Most of these methods use Meta heuristics algorithms, some of which are as follows:

Kazarlis et al.[7] Used the genetic algorithm (GA) to solve the timetable problem. kanoh and Sakamoto[8], tried to make new optimal tables based on the GA using a method called viralization and exploiting a knowledge base including past time tables. In other words, they attempted to use an almost correct answer that was used in the past. Sigl, Golub, and Mornar[9] considered the chromosome structure in their GA as a three-dimensional cube with the axes being time, day, and room cuts. In addition, after solving a problem with the GA to some extent Colorni, Dorigo, and Maniezzo[10], tried to optimize the table of professors using a function in order to reach a correct answer. Perzina[11] states that: “The timetable problem is known as a hard problem, thus there is no algorithm capable of solving this problem with the polynomial time complexity”. Neufeld et al.[12] exploited the following approach to convert the timetable problem into the graph coloring problem. In this conversion, each course is represented by a node of the graph. There is one edge between pairs of course that cannot be simultaneously scheduled. The unavailability and initial assignments can be managed by introducing some external constraints. The problem conversion is completed by assigning any time interval to a color.

In this paper, we want to use artificial fish-swarm algorithm to present a new method for solving the university exam timetabling problem.

2. Artificial Fish-Swarm Algorithm(AFSA)

AFSA is one of the swarm intelligence (SI) algorithms working based on population and random search. This algorithm was presented by Li Xiao Lei in 2002[13]. The basis of function of AFSA is derived from the social behavior of fish and works based on random search, population, and behaviorism. This algorithm includes characteristics like high convergence rate, sensitivity to the initial values of artificial fish, flexibility, and error tolerance, making it acceptable for solving optimization problems. AFSA has been widely used in numerous applications including clustering, resource leveling, proportional–integral–derivative controller (PID controller or three term controller), wide range, data mining, DNA coding sequences, etc. Figure 1 demonstrates the Pseudo-code of the AFSA.

Algorithm 1: Pseudo-code of AFSA

```
for each Artificial Fish i ∈ [1..N] do
    initialize x_i
Blackboard ← argmin f(x_i)
while stopping criterion is met do
    for each Artificial Fish i ∈ [1..N] do
        Perform Swarm Behavior on X_i(t) and Compute X_{i,swarm}
        Perform Follow Behavior on X_i(t) and Compute X_{i,follow}
        if f(X_{i,swarm}) ≥ f(X_{i,follow}) then
            X_i(t+1) ← X_{i,follow}
        else
            X_i(t+1) ← X_{i,swarm}
    if f(X_{Best-AF}) ≤ f(Blackboard) then
        Blackboard = X_{Best-AF}
```

Figure 1. Pseudo-code of the AFSA[13]

The AFSA function is based on the social behaviors of fish swarm in nature. In the underwater world, fish can find areas with higher amounts of food, which is achieved by fish individual or group search. Based on this trait, the artificial fish model has been proposed with free movement, food search, group movement, and following behaviors of fish. The objective function of the AFSA is the food density rate in the aqueous area. Finally, artificial fish reach a place with the highest density and concentration of food (global optimum). Table 2 shows the parameters of the AFSA.

Table 2. AFSA Parameters

Description	Formulation
Distance between positions X_i and X_j of two fish	$d_{ij} = \|\| x_i - x_j \|\|$
Individual artificial fish visibility	Visual
Artificial fish movement step	Step
Artificial fish crowd factor	δ

In this algorithm, the global optimum value can be obtained through the local search by each fish individually. The solution space is an environment to which the artificial fish belong. Moreover, each fish is aware of its environmental status, which is influenced by its own activities and activities of other fish.

Four main moves are considered for fish in the AFSA: the two following and grouped moves were considered as group learning, and food search and free moves as individual learning of artificial fish.

3. Proposed Method

In this study, a new method has been presented for university exams timetabling using the AFSA. The goal of the proposed method is to provide a program for university exams capable of meeting all the hard constraints as well as the highest number of soft constraints. In the following, the hard and soft constraints taken into account in the proposed method are stated.

Hard constraints: The most important constraints of providing the timetable for the exams are as follows:

- Exam interference(h_1): At a given time period, each student can have a maximum of one exam.
- Suitable room(h_2): Each exam must be held in an appropriate room.
- Room interference(h_3): At a given time period, a maximum of one exam can be held in each room.

Soft constraints: The most important soft constraints considered in the design of the proposed timetable for exams are as below:

- Students will have a maximum of one exam in a day(o_1).
- The interval between the exams of each student should be at least one day(o_2).

3.1 Coding

In the proposed AFSA algorithm, the current status of the artificial fish is X shown as the vector. To solve the university exams timetable problem, the current status of the fish is filled with the name of the course, place for holding the exam, and the schedule of the exam. For instance, if the number of courses is 7, one of the positions of the artificial fish can be as figure 2.; in this figure, 142, for example, indicates that the exam of the first course is held in class 4 and in the second schedule.

142	325	423	726	247	639	541

Figure 2. Current status of an artificial fish sample with seven courses

3.2 Objective Function

In the proposed algorithm, the objective function is calculated based on the constraints of the problem through formula 1. The lower the value of the objective function, the better the proposed solution. In this formula, m and Pn are the number of the violated constraints and penalty of this violation, respectively.

$$Fitness = \sum_{n=1}^{m}(Pn) \tag{1}$$

Table 3 represents the penalty value considered for each constraint violation. The greater the importance of complying with a constraint, the higher the amount of penalty considered for that constraint violation.

Table 3. Penalty Value Considered for each Constraint Violation

Constraint	Constraint violation penalty
h1	400
h2	1000
h3	1000
o1	5
o2	1

4. Simulation Results

The C # .Net 2017 programming language has been exploited to implement the algorithms. The programs have been run on a computer with a configuration of a Pentium Core i5 CPU and 8GB of RAM.

To compare the results of the proposed algorithm (AFSA) with other algorithms, the number of fish and iterations were considered to be 200 and 3000, respectively.

The standard benchmark of the Carter and ITC2007 have been used to evaluate the efficiency of the proposed algorithm. The standard Carter test data and ITC2007 test data contain 13 and 8 test data sets, respectively.

Table 4 illustrates the comparison of the results on the Carter benchmark. As it is clear, the proposed algorithm reaches better results in comparison with other algorithms in the three test data of ear-f-83 I, sta-f-83 I, and yor-f-83 I.

Table 4. Comparison of the results on the Carter benchmark

Data Set	SCHH [14]	SPHH [15]	GCHH [16]	GPHH [17]	Fish
car-f-92 I	3.93	4.32	4.28	4.00	3.97
car-s-91 I	4.50	4.97	4.97	4.62	4.56
ear-f-83 I	33.71	36.16	36.86	34.71	33.52
hec-s-92 I	10.83	11.61	11.85	10.68	10.87
kfu-s-93	13.82	15.02	14.62	13	13.40
lse-f-91	10.35	10.96	11.14	10.11	10.85
pur-s-93 I	–	–	4.73	4.8	4.78
rye-s-93	8.53	–	9.65	10.79	8.76
sta-f-83 I	151.52	161.91	158.33	158.02	151.36
tre-s-92	7.92	8.38	8.48	7.90	8.04
uta-s-92 I	3.14	3.36	3.4	3.12	3.13
ute-s-92	25.39	27.41	28.88	26	26.4
yor-f-83 I	36.53	40.77	40.74	36.2	35.8

Figure 3 demonstrates the comparison of the results on the Carter benchmark.

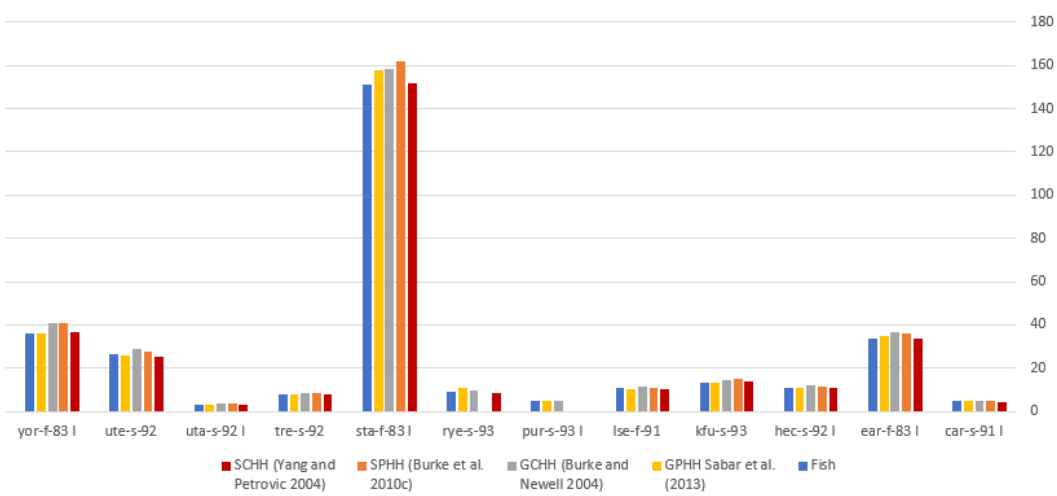

Figure 3. Comparison of the results on the Carter benchmark

Table 5 shows the comparison of the results on ITC2007 benchmark. As it can be observed, the proposed algorithm yields better solutions in the two Dataset 3 and Dataset 7 test data in comparison with other algorithms.

Table 5. Comparison of the results on the ITC2007 benchmark

Data Set	Pillay [18]	Sabar et al. [19]	Soghier and Qu [20]	Anwar et al. [21]	Fish
Dataset 1	8559	6234	5752	11823	6041
Dataset 2	830	395	1693	976	574
Dataset 3	11576	13002	14586	26770	11183
Dataset 4	21901	17940	21491	-	19847
Dataset 5	3969	3900	3844	6772	3926
Dataset 6	28340	27000	28480	30980	27625
Dataset 7	8167	6214	5182	11762	4925
Dataset 8	12658	8552	13711	16286	9718

Figure 4 demonstrates the comparison of the results on ITC2007 benchmark.

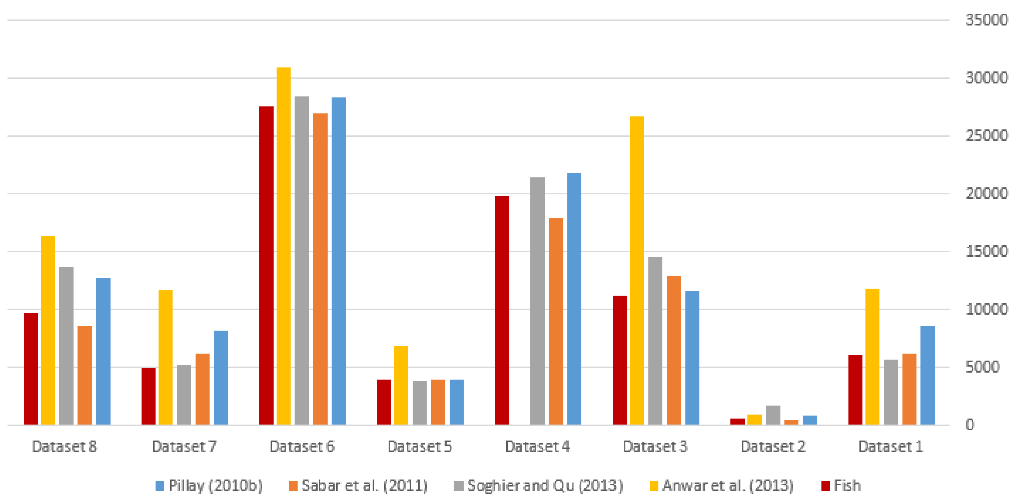

Figure 4. Comparison of the results on the ITC2007 benchmark

5. Conclusion

The problem of university exams timetabling is one of difficult problems, and numerous innovative methods have been developed to solve this problem. In order to schedule the university exams, hard constraints must be considered in order to design an acceptable timetable, in addition, soft constraints must also be taken into account in order to provide the desired timetable with high quality. In this paper, a new method for solving the university timetable problem has been presented using the AFSA. Two standard test data sets were used to evaluate the efficiency of the proposed method. The simulation results indicated that the proposed method in five test data from 22 test data led to better results compared to other algorithms. Therefore, it can be concluded that the proposed method is very effective in comparison with other methods.

References

[1] N. Pillay and R. Qu, "Examination Timetabling Problems," in *Hyper-Heuristics: Theory and Applications*: Springer, 2018, pp. 75-82.

[2] R. Gashgari, L. Alhashimi, R. Obaid, and T. Palaniswamy, "A Survey on Exam Scheduling Techniques," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 2018, pp. 1-5: IEEE.

[3] D. Btissam and R. Abounacer, "Multi-Objective examination Timetabling Problem: Modeling and resolution using a based ε-constraint method," *IJCSNS*, vol. 1 ,٧no. 4, p. 192, 2017.

[4] K.-H. Krempels and A. Panchenko, "An approach for automated surgery scheduling," in *Proceedings of the sixth international conference on the practice and theory of automated Timetabling*, 2006, pp. 209-233.

[5] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Computers & Operations Research,* vol. 54, pp. 142-154, 2015.

[6] N. Leite, C. M. Fernandes, F. Melício, and A. C. Rosa, "A cellular memetic algorithm for the examination timetabling problem," *Computers & Operations Research,* vol. 94, pp. 118-138, 2018.

[7] S. Kazarlis, V. Petridis, and P. Fragkou, "Solving university timetabling problems using advanced genetic algorithms," *GAs,* vol. 2, no. 7, pp. 8-12, 2005.

[8] H. Kanoh and Y. Sakamoto, "Knowledge-based genetic algorithm for university course timetabling problems," *International Journal of Knowledge-based and Intelligent Engineering Systems,* vol. 12, no. 4, pp. 283-294, 2008.

[9] B. Sigl, M. Golub, and V. Mornar, "Solving timetable scheduling problem using genetic algorithms," in *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, 2003, pp. 519-524: IEEE.

[10] A. Colorni, M. Dorigo, and V. Maniezzo, "A genetic algorithm to solve the timetable problem," *Politecnico di Milano, Milan, Italy TR,* pp. 90-060, 1992.

[11] R. Perzina, "Solving the university timetabling problem with optimized enrollment of students by a self-adaptive genetic algorithm," in *International Conference on the Practice and Theory of Automated Timetabling*, 2006, pp. 248-263: Springer.

[12] G. Neufeld and J. Tartar, "Graph coloring conditions for the existence of solutions to the timetable problem," *Communications of the ACM,* vol. 17, no. 8, pp. 450-453.١٩٧٤ ,

[13] L. Xiao, "lei, SHAO Zhi\| jiang, QIAN Ji\| xin (Institute of Systems Engineering, Zhejiang University, Hangzhou 310027, China); An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm [J]," *Systems Engineering-theory & Practice* ,vol. 11, 2002.

[14] Y. Yang and S. Petrovic, "A novel similarity measure for heuristic selection in examination timetabling," in *International Conference on the Practice and Theory of Automated Timetabling*, 2004, pp. 247-269: Springer.

[15] E. K. Burke ,R. Qu, and A. Soghier, "Adaptive selection of heuristics for improving exam timetables," *Annals of Operations Research,* vol. 218, no. 1, pp. 129-145, 2014.

[16] E. K. Burke and J. P. Newall, "Solving examination timetabling problems through adaption of heuristic orderings," *Annals of operations Research,* vol. 129, no. 1-4, pp. 107-134, 2004.

[17] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Grammatical evolution hyper-heuristic for combinatorial optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 840-861, 2013.

[18] N. Pillay, "Evolving hyper-heuristics for a highly constrained examination timetabling problem," in *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT'10)*, 2010, pp. 336-346.

[19] N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Applied Intelligence*, vol. 37, no. 1, pp. 1-11, 2012.

[20] A. Soghier and R. Qu, "Adaptive selection of heuristics for assigning time slots and rooms in exam timetables," *Applied Intelligence*, vol. 39, no. 2, pp. 438-450, 2013.

[21] K. Anwar, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "Harmony search-based hyper-heuristic for examination timetabling," in *2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, 2013, pp. 176-181: IEEE.