Comparing the Methods of Creating Educational Timetable

Vahideh Sahargahi[1], Mohammadreza Feizi Drakhshi[2]
[1]M.Sc. in Artificial Intelligence, Faculty of Electronic and Computer, Tabriz University, Tabriz, Iran
[2]Assistant Professor, PhD in Artificial Intelligenc, Faculty of Electronic and Computer, Tabriz University, Tabriz, Iran

Abstract
The problem of creating educational timetable is one of the NP-Complete problems to which many solutions have been introduced in various studies. Nowadays there are intelligent methods based on evolutionary algorithms which are used for timetabling problems. Some of such methods are sectionicle swarm optimization (PSO), the memetic algorithm, fuzzy systems, the harmony search, and ant colony optimization. One of the most important challenges, dealt with in this paper, is to determine which one of these methods could be highly efficient to create an educational timetable. For this purpose, different concepts such as limitations, methods of showing timetables, and databases with various sizes and properties should be considered. These concepts were comprehensively introduced and discussed in this paper. There are many barriers to the accurate comparison of methods for creating educational timetables. Dealing with such problems, this paper compared some of these methods briefly. Given the previous studies, it can be stated that hybrid methods are more efficient than not hybrid ones. Comparing different methods, it is worth mentioning that the memetic algorithm and PSO were more efficient than the genetic algorithm while honey bee mating optimization algorithm indicated a better performance than the genetic, memetic, and ant colony optimization algorithms. Moreover, the harmony search method was more efficient than most of the abovementioned methods such as ant colony optimization, fuzzy method, the memetic and genetic algorithms.

Keywords:
Educational Timetable, Evolutionary Algorithms, Hard and Soft Limitations

1. Introduction

The problem of creating educational timetable is one of the important requirements at schools, universities and education centers. It is classified as NP-Complete problems. Many solutions and studies have been proposed for this problem. The timetabling problem refers to the allocation of events to predefined time periods in a way that some limitations are applied to these allocations. These limitations fall into two groups of soft and hard. Hard limitations are the ones which must be considered to present a timetable. On the other hand, soft limitations refer to the ones which can be ignored; however, they should be applied as much as possible. The quality of a timetable is evaluated with the consideration of soft limitations which were applied to it. New approaches which are used to configure timetabling problems are evolutionary algorithms. They fall into several groups including the hill climbing search, the firefly method, the genetic algorithm, and so on. The genetic algorithm is one of the strongest and most applicable algorithms in search and optimization problems. Given the large amount of information, using the random method in educational timetabling problems for university courses will increase the time of finding the optimal answer. This paper was intended to analyze and compare different methods for educational timetabling. In Section 2, the basic concepts of educational timetabling are presented. The available optimization methods for timetabling are discussed in Section 3. The limitations and solutions to the comparison of these methods are stated in Section 4. The available methods are compared with respect to database in Section 5. Finally, the conclusion and future works are presented in Section 6.

2. Basic Concepts

In this section, the basic concepts of educational timetabling problems are dealt with. These concepts include limitations, methods of showing timetables, and databases with different sizes and properties. They are discussed in details in the following.

2-1- Timetable

The problem of timetables refers to the process of allocating limited locational-temporal resources to a number of events under the condition of meeting a large numbers of limitations. Although this definition includes a very broad range of problems, this paper dealt with the problem of timetables for classes or examinations.

2-2- Describing Educational Timetabling Problem

The university timetabling problem especially educational and examinational timetables are difficult tasks which education centers deal with. It is a resource allocation problem with many limitations. In this problem, a number of course groups are defined with a number of sessions, times and locations of which should be determined with respect to variety of limitations.

Manuscript received December 5, 2016
Manuscript revised December 20, 2016

2-3- Different Types of Limitations in Educational Timetabling Problem

Limitations fall into two groups of soft and hard. Hard limitations refer to the ones which must be followed to present a timetable. On the contrary, soft limitations are those which can be ignored; however, they should be applied as much as possible. The quality of a timetable is evaluated with respect to the soft limitations applied to it.

2-3-1- Hard Limitations

Some of the hard limitations are as follows:

- Each teacher can only teach one course in each time period.
- Each course can only be taught by one teacher in one time period.
- Students can select one course in one time and sectionicipate in its class.
- Each classroom can only be allocated to one course in each time for being held.
- Some of the courses must be taught in specific classrooms such as laboratory courses.
- Each teacher’s working hours are limited during the week.
- If more than one course is taught by the same teacher, they cannot occur in the same time period.
- Some courses which need the same class cannot be put in one time slice.
- Some of the classes must be held in special hours during the week.

2-3-2- Soft Limitations

Some of the soft limitations can be as follows:

- Courses with similar topics should be normally distributed among week hours.
- A class should be selected near the teacher’s office.
- If possible, the classroom considered for the courses taught by one teacher should not change.
- It is better to present some of the courses in specific hours on sectionicular days of week.
- The number of rooms used for classes should be minimized.
- The class capacity should be taken into account.
- The priorities of each teacher for selecting each course and weekly time slices should be taken into consideration.

2-4- Definitions in the Educational Timetabling Problem

- Event: It is an activity which should be timetabled.
- Resource: Resources are rooms, laboratories and other items which are required by events.
- Time Slice: It is the time period in which the events must be timetabled.
- People: They are students and teachers who exist along with events.
- Collision: It happens when more than one event collide with each other like timetabling more than one course in a specific time period.

2-5- Methods of Showing Educational Timetabling Problem

A timetable is indicated as a 2D matrix in which rows refer to classes while columns represent time periods. Therefore, the class i in the time period j in the matrix R indicates a specific event whose number is n (rij=n). Asection from that, an array of events which were not timetabled are kept for each timetable. This method applies all the limitations pertaining to the problem. Therefore, one event can be placed in a timetable when all of the hard limitations were taken into consideration after adding it. The number of allocated time periods is considered to be equal to the number of useful time periods plus the constant ones. To decrease the complexity of problem, for example to consider 40 hours of timetable to be a sectionicle, 20 times will be determined, and 8 hours of class can be set in 4 times. Therefore, 20 times will exist during the week. The 2D view of an educational timetable is indicated in Figure 1.
Moreover, the educational timetabling can be shown as a 3D matrix in a way that the first dimension refers to the number of classes while the second dimension represents the number of weekdays, and the third dimension indicates the time periods on each day. Figure 2 indicates the 3D view of educational timetable.

	Saturday 8-9	Sunday 9-10	…	Monday 13-14	…	Wednsday 17-18	Wednsday 18-19
Class 1	81	24	…	56	…	9	89
Class 2	76	12	…	53	…	39	69
…	29	59	…	20	…	86	75
Class n	18	42	…	15	…	33	5
Lab 1	89	9	…	96	…	72	49
…	41	30	…	78	…	57	86
Lab n	79	60	…	71	…	10	80

Figure 1: The 2D View of Educational Timetable [1]

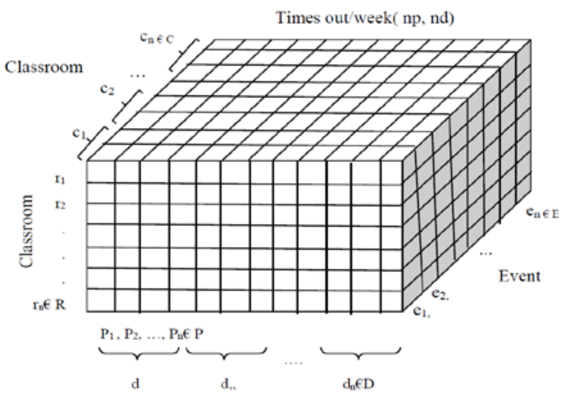

Figure 2: The 3D View of Educational Timetable [1]

2. 6 Standard Databases for Educational Timetabling

In this section, some of the standard databases, which are used by researchers to evaluate the proposed methods, are introduced.

2-6-1- Database 2002

This database can be seen in the following table including 4 columns, the first of which shows the names of events and resources stated in the database. The columns 2 to 4 indicate the number of events and resources in three sizes including small, medium and large, respectively. The information pertaining to the different sizes of this database is available in the table.

Table 1: Database 2006 [12]

Features \ Categories	Small	Medium	Large
Number of courses	100	400	400
Number of rooms	5	10	10
Number of features	5	5	10
Number of students	80	200	400
Max. courses per student	20	20	20
Max. students per course	20	50	100
Approx. features per room	3	3	5
Percent feature use	70	80	90

Table 2: The Events and Resources in the Different Sizes of Database [12]

Features \ Categories	#Events	#Rooms	#Students	#Features
Small 1	100	5	80	5
Small 2	100	5	80	5
Small 3	100	5	80	5
Small 4	100	5	80	5
Small 5	100	5	80	5
Medium 1	400	10	200	5
Medium 2	400	10	200	5
Medium 3	400	10	200	5
Medium 4	400	10	200	5
Medium 5	400	10	200	5
Large	400	10	400	10

In Paper [12], 24 different methods which used Database 2002 to create timetables were compared with each other. For small data, the majority of methods could obtain the desirable result at the highest quality. Among the methods which were compared, the tabu search method based on the memetic algorithm could obtain the best result for medium databases. The graph coloring method achieved an appropriate result for large data.

2-6-2- ITC 2002

The properties of Database ITC 2002[1] can be seen in Table 3 including seven columns, first of which indicates a list of events and resources. The other columns represent the number of events, resources and rooms allocated to events, events allocated to students, and students allocated to events.

Table 3: Database ITC 2002 [12]

	#Events	#students	#Rooms	Rooms/events	Events/students	Students/events
COMP01	400	200	10	1.94	17.75	8.88
COMP02	400	200	10	1.92	17.23	8.62
COMP03	400	200	10	3.42	17.70	8.85
COMP04	400	300	10	2.45	17.43	13.07
COMP05	350	300	10	1.78	17.78	15.24
COMP06	350	300	10	3.59	17.77	15.23
COMP07	350	350	10	2.87	17.48	17.48
COMP08	400	250	10	2.93	17.58	10.99
COMP09	400	220	11	2.58	17.36	8.68
COMP10	400	200	10	3.49	17.78	8.89
COMP11	400	220	10	2.06	17.41	9.58
COMP12	400	200	10	1.96	17.57	8.79
COMP13	400	250	10	2.43	17.69	11.05
COMP14	350	350	10	3.08	17.12	17.12
COMP15	350	300	10	2.19	17.58	15.07
COMP16	440	220	11	3.17	17.75	8.88
COMP17	350	300	10	1.11	17.67	15.45
COMP18	400	200	10	1.75	17.56	8.78
COMP19	400	300	10	1.94	17.71	13.28
COMP20	350	300	10	3.43	17.49	14.99

2-6-3- ITC 2007

Table 4 is associated with Database ITC 2007. The first column indicates the classification name of database. Other columns represent the number of students, classrooms, events, and other properties pertaining to the database.

Table 4: Database ITC 2007 [12]

	#Events	#Rooms	#Feature	#Students	Max student per event	Max event per students	Max features per room	Max features per event
1	400	10	10	500	33	25	3	1
2	400	10	10	500	32	24	4	2
3	200	20	10	1000	98	15	3	2
4	200	20	10	1000	82	15	3	2
5	400	20	20	300	19	23	2	1
6	400	20	20	300	20	24	3	2
7	200	20	20	500	43	15	5	3
8	200	20	20	500	39	15	4	3
9	400	10	20	500	34	24	3	1
10	400	10	20	500	32	23	3	2
11	200	10	10	1000	88	15	3	1
12	200	10	10	1000	81	15	4	23
13	400	20	10	300	20	24	2	1
14	400	20	10	300	20	24	3	1
15	200	10	20	500	41	15	2	3
16	200	10	20	500	40	15	5	3
17	100	10	10	500	195	23	4	2
18	200	10	10	500	65	23	4	2
19	300	10	10	1000	55	14	3	1
20	400	10	10	1000	40	15	3	1
21	500	20	20	300	16	23	3	1
22	600	20	20	500	22	25	3	2
23	400	20	30	1000	69	24	5	3
24	400	20	30	1000	41	15	5	3

1 http://www.idsia.ch/Files/ttcomp200

2-7- The Criteria for Evaluating Timetables of Universities

In various papers, different evaluation criteria were stated to assess the quality of timetables. One of the methods used to evaluate timetables is to consider the events which were not timetabled [2]. The number of students for whom soft limitations were followed is another method which can be used to evaluate the quality of timetables. In some papers, the number of executions were used to obtain an appropriate timetable to evaluate the methods [3].

3. The Available Optimization Methods for Designing Timetables

Different optimization methods were used by researchers to present educational timetables.
These methods con be grouped as following [25], graph coloring, clustering methods, constraint based methods (e. g. integer programing), meta heuristic methods (e. g. genetic algorithms, simulated annealing, tabu search and ant colony). In Paper [21], various methods were stated to present educational timetables for schools and the limitations considered for each of them. In Paper [20], researchers stated different methods of proposing educational timetables for universities, presenting a timetable for examinations, and classifying students. Some of the methods used to present educational timetables include PSO [1] [6] [7], the memetic algorithm, [2], fuzzy logic [3], the harmony search algorithm [5], and ant colony optimization [4] [15] [16]. Some of such methods are explained in details here.

3-1- PSO Algorithm

The particle swarm optimization algorithm, which is also known as the birds' algorithm, is included in the family of cumulative intelligence methods. It is one of the successful algorithms used for continuous and discrete optimization. The Hybrid Particle Swarm Optimization (HPSO) is the improved version of PSO introduced in Paper [1] for the educational timetabling problem. In this method, the table of teachers and teaching is considered to be the Particle. Some of the characteristics of HPSO include using a correctional process for impossible tables which cannot be scheduled and applying the local search to look for the best location for particle due to the fact the educational timetabling problem is discrete. In this paper, the soft limitations which were not taken into account in other papers were stated, a fact which is one of the advantages of their proposed methods. In the mentioned soft limitations, teachers and students can state their priorities to select educational timetabling in a table in order to see what time periods they prefer to have classes on which days. Teachers can introduce the most selections of time periods for their spare time. In fact, they declare the times they do not like to have classes in. Teachers can state their opinions on how and when the classes are held. This method minimizes the movement of students between classes (shortening the distance between successive classes which are held for one student). Since the willingness of each teacher towards teaching different courses is clear, each teacher specifies the priorities based on his or her preferences. The main purpose is to find a flexible timetable to which all the hard limitations can be applied, and the Total Priority Values (TPV) can be maximized for each teacher and class in it. The fitness function and evaluation criteria were defined with respect to priority values pertaining to teachers and classes. The HPSO method was compared to the genetic algorithm with respect to the same fitness function, and the evaluation results indicated its high efficiency.
The Constriction Particle Swarm Optimization (CPSO) was introduced in [6] to design the educational timetables. Using a heuristic exchange function, this method is able to create educational timetables with respect to the request of teachers, classes and hard and soft limitations. The local search exercised in this method to create the educational timetable increased teachers' satisfaction and the quality of obtained solutions in addition to fixing the problem of premature convergence in PSO. In Paper [7], the genetic algorithm was compared to PSO. The results indicated that PSO needed fewer repetitions than the genetic algorithm did to obtain the optimal answer. The penalty value of PSO was lower than that of the genetic algorithm, too.

3-2- The Memetic Algorithm

The classic genetic algorithms found the answer areas at a good speed; however, they spent a great deal of time obtaining an answer with the desired accuracy. This flaw can be improved to some extent by using the knowledge in the problem or adding the local search phase to the evolutionary cycle. Being inspired by the idea of meme, introduced by Richard Dawkins, researchers proposed algorithms in which memes, as local searchers, improved the obtained answers so that the search process would be quicker and more efficient. Such methods are called the memetic or cross algorithms. Regarding adaptability, such algorithms fall into three classes of static, comparative, and self-comparative. In the first two groups, the memes are not changed in the search process, and the only important problem is to select the best meme. However, in the third process, the memes are evolved in a separate population and adapted to the problem in an evolutionary way.
The memetic algorithm was proposed to create the educational timetables in [1]. One of the methods which can be used to evaluate timetables is to consider the number of events which were not timetabled. Using this criteria as the fitness function is not able to tell the difference between the

solutions which are very similar to each other, but are structurally different. Therefore, another fitness function should be used to tell the difference between solutions. The fitness function used in this paper had some characteristics such as using the experiences to learn, low computational costs, and considering the structure of solution. When the number of samples is small or medium, this method indicated more efficiency than other heuristic methods such as the genetic algorithm, H, and GGA. After the memetic algorithm, the H algorithm had better results, and then HAS and GGA were in the next places.

3-3- The Ant Colony Optimization Algorithm

The ant colony optimization algorithm was proposed by Thepphakorn et al. in Paper [4] to create the educational timetable. Moreover, two new types of this algorithm named Best-Worst Ant System (BWAS) and Best-Worst Ant Colony System (BWACS) were added to ACO. Then the strategy of local search was added to them to improve the efficiency of these two proposed methods and find the best timetable. Adding the local search increased the efficiency of the proposed method up to 40%; however, it increased the runtime, too.

The efficiencies of BWAS and BWACS were higher than that of ACO. Comparing the two proposed methods, it can be stated that BWACS is appropriate for large problems, and BWAS is good for small problems.

In Paper [11], Ayob et al. used the hybrid of two different methods with ACO for educational timetabling. The hybrid methods were named ACS-TS, combining ant colony system and tabu search, and ACS-SA, resulting from the combination of ant colony system and the simulated annealing. The evaluation results indicated that both the hybrid methods had better efficiencies with higher qualities of answers in comparison with ant colony system. Moreover, the authors of this paper believed that their hybrid method can be used for other optimization problems.

In Paper [15], Nothegger et al. proposed a method based on ant colony system to create the educational timetable. The most important characteristic of their method was to use two separate but very simple matrices for convergence. Furthermore, with the parallel implementation of the program, they could improve the quality of the proposed method. Database ITC 2007 was used for evaluation. The evaluation results indicated optimal tables for different sized of the abovementioned database in the proposed method. In the next step, they added the local search in order to increase the efficiency of ant colony system. The evaluation results indicated the efficiency increase more with the local search.

Two methods of ant colony system and simulated annealing algorithm were investigated and compared to create the examinational timetables by Chmait et al. According to their results, ant colony system was more efficient than the simulated annealing algorithm in terms of the runtime because the latter needed more time to discover and evaluate the neighbors in the different periods of program. Regarding the cost, the ant colony system had lower costs in most cases compared to the ant colony system in which if the number of ants go up, the runtime increases. However, it does not increase the quality of tables considerably. Moreover, if the number of ants decreases too much, the runtime is reduced; however, a table of low cost cannot be achieved [16].

3-4- Fuzzy Logic

BSC is one of the old methods to create educational timetables. In this method, the events are sorted with respect to their hardness. Given the fact that what criterion was used for sorting, this method can be done in different ways. The authors of [3] proposed the use of fuzzy logic to sort events in BSC. The fuzzy system, used in this paper, had 3 inputs and one output. The inputs include Saturation Degree (SD), Largest Enrollment (LE) and Largest Degree (LD). The output of this system will be the hardness level of events which is used in BSC. The inference system used in this method was Mamdani, and the maximum function was used to attach the rules. A function based on centrality was used to for deactivation. The fitness criterion used to calculate the quality of the proposed timetable was the number of students for whom soft limitations were followed. The evaluations were done on 10 databases of different sizes.

The results of evaluations done in [3] indicated that the fuzzy method had a higher quality in presenting educational timetables with BSC in comparison with not hybrid methods. Moreover, the number of times in which rescheduling is needed by the algorithm to find the answer was calculated for different methods. According to the results, the fuzzy method needed fewer times of rescheduling for more databases. Among other not hybrid methods, the SD method needed the fewer number of rescheduling to find the timetable. However, this method could not find an appropriate answer for large databases. The advantages of using fuzzy method to calculate the hardness of events in BSC algorithm include their efficiency while encountering large databases.

3-4-1- The Ant-Fuzzy Method

In Paper [24], the fuzzy theory was used to increase the efficiency of ants' populations. The educational timetabling problem is like a 2D matrix in which rows refer to classrooms and time periods. Its columns represent the events. This matrix is indicated as the following graph. The evaluation results indicated the high efficiency this algorithm in complicated problems with various limitations.

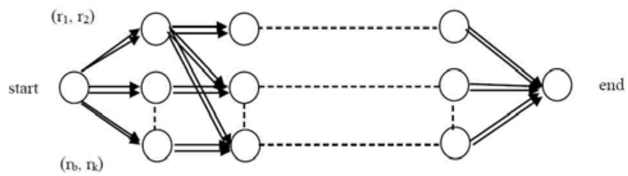

Figure 2: The Educational Timetabling Graph in Which Each Row Indicates a Pair of Classroom and Time Period While Each Column Represents an Event [24]

3-5- The Harmony Search Method

In Paper [10], Azmi et al. used the Adapted Harmony Search Algorithm (AHSA) for the educational timetabling problem and also proposed the linked harmony search method to create the educational timetable. In this method, the hill climbing algorithm was used to improve the local search, and PSO was employed to improve the public search and increase the quality of timetables.

AHSA and HHSA were evaluated on databases with different sizes. These methods were compared with 26 other methods which used the same database. The characteristics pertaining to the compared methods can be seen in Table 5. The results of comparisons can also be observed in this table. AHSA is able to find appropriate solutions for small and medium databases. In comparison with RII, RRLS, MMAS, VNS, GHH, FMHO, HEA, and THH, the proposed method had better results than GHH and FMHO for small databases. Considering medium databases, it also had better results than VSN and some cases of FMHO [5].

Table 5: The Methods Compared with the Harmony Search Algorithm

Alghorithm Type	No. key	Method
HAS Based	HHSA AHSA MHSA HAS-MPAR	Hybrid Harmony Search Algorithm Adapted Harmony Search Algorithm Modified Harmony Search Algorithm Harmony Search Algorithm With Multi Pitch Adjusting rate
Huristic Based	RRLS MMAS THH VNS FMHO DCFHH GHH RII GD NGD NNGDHH-SM NNGDHH-DM GDTS PCA LARD	Random restart Local Search Max-Min Ant System Tabu-Search Hyper-Heuristic Variable Neighboorhood Search Fuzzy Multiple Heuristic Ordering Destributed Choice Function Hyperhuristic Graph-based Hyper-Huristic Randomized Iterative Improvement Great Deluge Non-Linear Great Deluge Non-Linear Great Deluge Hyper-Huristic Static Memory Non-Linear Great Deluge Hyper-Huristic Dynamic Memory Great Deluge and tabue Search Particle Collision Algorithm Late Acceptance Randomized
Memetic Based	1. EGD 2. HEA 3. MA 4. ENGD 5. EMGD 6. GGA 7. ACS-SA 8. ACS-TA 9. GAGLS	Extended Great Deluge Hybried Evolutionary Approach Memetic Algorithm Evolutionery Non-Linear Great Deluge Electomagneticm Mechanism Great Deluge Guided Genetic Algorithm Ant Conoly system with Simulated Annealing Ant Colony System with tabu Search Genetic Algorithm with Guided and Local Search

Table 6: Comparing the Harmony Search Methods with Other Available Methods for Creating Timetables [10]

Algorithm Type	No.	Key	Small 1	Small 2	Small 3	Small 4	Small 5	Medium	Medium	Medium	Medium	Medium	Large
HSA based	1.	HHSA(*best*)	0	0	0	0	0	99	73	130	105	53	385
	2.	AHSA(*best*)	3	2	4	3	0	223	216	272	202	177	–
	3.	MHSA(*best*)	0	0	0	0	0	168	160	176	144	71	417
	4.	HSA-MPAR(*best*)	0	0	0	0	0	124	117	148	132	67	424
Heuristic based	5.	RRLS *(avg.)*	8	11	8	7	5	199	202.5	–	177.5	–	–
	6.	MMAS *(avg.)*	1	3	1	1	0	195	184	284	164.5	219.5	851.5
	7.	THH *(best)*	1	2	0	1	0	146	173	267	169	303	1166
	8.	VNS *(best)*	0	0	0	0	0	317	313	375	247	292	–
	9.	FMHO *(best)*	10	9	7	17	7	243	325	249	285	132	1138
	10.	DCFHH *(best)*	1	3	1	1	0	182	164	250	168	222	–
	11.	GHH *(best)*	6	7	3	3	4	372	419	359	348	171	1068
	12.	RII *(best)*	0	0	0	0	0	242	161	265	181	151	–
	13.	GD *(best)*	17	15	24	21	5	201	190	229	154	222	1066
	14.	NGD *(best)*	3	4	6	6	0	140	130	189	112	141	876
	15.	NGDHH-SM *(best)*	0	0	0	0	0	71	82	137	55	106	777
	16.	NGDHH-DM *(best)*	0	0	0	0	0	88	88	112	84	103	915
	17.	GDTS *(best)*	0	0	0	0	0	78	92	135	75	68	556
	18.	PCA *(best)*	1	1	1	1	0	136	138	165	143	135	789
	19.	LARD *(best)*	0	0	0	0	0	149	132	200	138	173	855
Memetic based	20.	EGD *(best)*	0	0	0	0	0	80	105	139	88	88	730
	21.	HEA *(best)*	0	0	0	0	0	221	147	246	165	130	529
	22.	MA *(best)*	0	0	0	0	0	227	180	235	142	200	–
	23.	ENGD *(best)*	0	1	0	0	0	126	123	185	116	129	821
	24.	EMGD *(best)*	0	0	0	0	0	96	96	135	79	87	683
	25.	GGA *(best)*	0	0	0	0	0	240	160	242	158	124	801
	26.	ACS-SA *(best)*	0	0	0	0	0	117	121	158	124	134	645
	27.	ACS-TS *(best)*	0	0	0	0	0	150	179	183	140	152	750
	28.	GAGLS *(best)*	0	0	0	0	0	139	92	122	98	116	615

HHSA produced the best possible results for each of the five small databases. However, AHSA could not obtain ideal results for small data. Seventeen other methods (GDTS, LARD, EGD, HEA, MA, EMDG, GGA, ACS-SA, ACS-TS, MHSA, HAS-MPAR, VNS, RII, NGDHH-SM, GAGLS, and NGDHH-SM) obtained the best possible results in small data like the adapted search algorithm.

Among five medium data, HHSA could generate tables with better qualities for the two databases in comparison with other methods. NGDHH-SM and NGDHH-DM could obtain desirable results for medium data. Moreover, HHSA could achieve a table with higher quality databases including large data in comparison with other methods [10].

3-6- Honey-Bee Mating Optimization Algorithm

The improved version of Honey-Bee Mating Optimization for educational timetabling problem (HBMO-ETP) was proposed for courses and examination by the authors of Paper [17]. Their proposed method was compared with the honey bee optimization algorithm and other similar methods with the consideration of the benchmark database. The evaluation results indicated its high efficiency in some samples. The honey bee algorithm have premature convergence because the initial population is not updated and improved during the search process; however, the improved honey bee algorithm indicated better efficiency due to the exploration and exploitation of the search space. Some of the disadvantages of HBMO-ETP, compared with other heuristic methods, may include the fact that there are different parameters which should be initiated with appropriate values like the population size of honey bees, the population size of seeds, and so on.

In Paper [18], HBMO algorithm was used for the examinational timetabling problem. Each honey bee explores the search space on a forward path and then shares its information with other honey bees in the hive on the way back. On the way forward, this algorithm uses other local search algorithms such as the hill climbing algorithm and the simulated annealing algorithm. Three methods (tournament, rating, and disruptive selection strategy) were proposed for the selection of the new working bee in order to create variety on the way back. The disruptive selection strategy had higher efficiency in comparison with tournament and rating methods. Moreover, a new method was proposed for the selection of neighbors' structures in order to improve the neighbor search. Keeping the structures of useful neighbors on the list of winners and leading the search, this mechanism would increase the efficiency of the proposed method.

4. Limitations and Available Solutions to the Comparison of Educational Timetabling Algorithms

Some of the limitations to the comparison of various methods proposed by different researchers on educational timetabling can be as follows:

- The variety of soft and hard limitations considered by researchers
- The variety of evaluation criteria for the proposed methods
- Difference in the ways timetabling problem is indicated
- Difference in the databases considered
- The hardware differences of the systems used for evaluation
- Difference in the optimization algorithms used for presenting timetables

These limitations make it hard to compare the theories of different methods proposed by researchers. For the logical comparison of different methods, all the circumstances and parameters of a problem should be considered the same, and the difference should only be based on the comparison criterion. Therefore, different methods should be evaluated and compared under similar conditions. In the next section, the results of comparing some papers which used this method to compare different educational timetabling methods are stated.

Another method used to compare these algorithms correctly and logically is to present the timetable resulting from various methods for different databases and to compare them with the consideration of soft and hard limitations. Unfortunately, researchers avoided presenting the timetable resulting from the evaluations they did in the published papers, only the website [23] was designed to do this kind of comparison for Database ITC 2007. On this website, the researchers compete with each other by sending the timetables resulting from their own proposed methods. The best methods were stated in Section 5-3 for the different sizes of this database.

According to Table 7, generally speaking, it can be stated that hybrid methods are more efficient than not hybrid methods. If we want to compare the proposed methods briefly, it can be discussed that the memetic algorithm and PSO were more efficient than the genetic algorithm [7]. HBMO algorithm had better results in creating timetables rather than the genetic, memetic, and ant colony optimization algorithms [17]. The harmony search method was the most efficient one compared with all of the proposed methods like ant colony, fuzzy, memetic, and genetic [10] [5].

Table 7: Comparing Hybrid Methods with Not hybrid Methods

Refrence s	Not Hybrid	Fitness Average	Hybrid	Fitness Average
[22]	ABC	312.6454	INMGD-ABC (Great Deluge, Artificial Bee Colony Algorithm)	85.75
[17]	HBMO	26.3977	HBMO-ETP (Honey-Bee Mating Optimization, Hybrid Graph Coloring Heuristics)	25.6385
[18]	Self-Adaptive DBCO	10558	Self-Adaptive DBCOSA (Bee Colony Optimization Based On Disruptive Selection, Simulated Annealing)	9826
[18]	Self-Adaptive DBCO	10558	Self-Adaptive DBCOLAHC (Bee Colony Optimization Based On Disruptive Selection, Late Acceptance Hill Climbing Algorithm)	9318
[11]	MMAS	169.8181	ACS-SA (Ant Colony Systems, Simulated Annealing)	118.0909
[11]	MMAS	1698181	ACS-TS (Ant Colony Systems, Tabu Search)	141.2727
[13]	GALS	582.7000	NHA (Local Search And Using Event Based Grouping Students)	580.5000

5. Comparing the Available Methods Based on Databases with Different Sizes

5-1- Database 2002

In Paper [12], 24 different methods which used Database 2002 to create timetables were compared with each other. For small data, the majority of methods could obtain the best result with the highest possible quality. Among the methods which were compared, tabu search, based on the memetic algorithm, could have the best results for medium databases. The graph coloring method obtained an appropriate result for large data.

5-2- ITC 2007

The timetables pertaining to different methods using Database ITC 2007 were compared on the website [23]. The best method was introduced for different sizes of databases.

Table 8: Introducing the Best Methods for Creating Timetables on Database ITC 2007 [23]

Instance Method used	Method used
comp01	Tabu Search
comp02	SAT-Modulo-Theory
comp03	Local Search
comp04	Local Search
comp05	Local Search
comp06	SAT-Modulo-Theory
comp07	SAT-Modulo-Theory
comp08	Hybridised Algorithm
comp09	Tabu Search
comp10	SAT-Modulo-Theory
comp11	Tabu Search
comp12	Tabu Search
comp13	Tabu Search
comp14	Mathematical Programming
comp15	Tabu Search
comp16	SAT-Modulo-Theory
comp17	SAT-Modulo-Theory
comp18	Tabu Search
comp19	Local Search
comp20	SAT-Modulo-Theory
comp21	SAT-Modulo-Theory

5-3- Socha 2002

In Paper [12], the 24 methods which used the abovementioned databases for timetabling were compared with each other. Table 9 indicates the results of evaluations. Methods of A_4, A_5, A_7, A_{11}, and A_{13} are inappropriate for small data. Methods like A_{10} and A_2-A_{14} obtained the best results for medium and large data, respectively.

Table 9: Comparing 24 Methods for Creating Timetables on Database 2002 [12]

Large	Medium5	Medium4	Medium3	Medium2	Medium1	Small5	Small4	Small3	Small2	Small1	Datasets Algorithms
851.5	219.5	164.5	248	184	195	0	1	1	3	1	A_1
1166	303	169	267	173	146	0	1	0	2	1	A_2
1138	132	285	249	325	243	7	17	7	9	10	A_3
932	292	247	357	313	317	0	0	0	0	0	A_4
757	151	181	265	161	242	0	0	0	0	0	A_5
1068	171	348	359	419	372	4	3	3	7	6	A_6
529	130	165	246	147	221	0	0	0	0	0	A_7
1066	222	154	229	190	201	5	21	24	15	17	A_8
876	141	112	189	130	140	0	6	6	4	3	A_9
851.5	219.5	164.5	248	184	195	0	1	1	3	1	A_{10}
1027	276	321	251	258	254	4	0	2	4	2	A_{11}
730	88	88	139	105	80	0	0	0	0	0	A_{12}
912	190	149	216	197	175	0	0	0	0	0	A_{13}
798	166	148	191	154	176	0	3	3	5	5	A_{14}
627	101	69	133	73	77	0	0	0	0	0	A_{15}
653	61	32	102	70	50	0	0	0	0	0	A_{16}
980	98	103	149	98	93	0	0	0	0	0	A_{17}
424	73	132	190	117	124	0	0	0	0	0	A_{18}
851.5	171	164.5	248	173	146	0	1	0	2	1	A_{19}
768	79	86	158	102	99	0	0	0	0	0	A_{20}
801	124	158	242	160	240	0	0	0	0	0	A_{21}
555	64	67	115	77	70	0	0	0	0	0	A_{22}
786	135	143	165	138	136	0	1	1	1	1	A_{23}
589	160	75	135	108	117	0	0	0	0	0	A_{24}

5-4- Ben Paechter Database

In Paper [12], researchers compared 15 methods which used Database Ben Paechter. The results can be seen in Table 10

Table 10: Comparing the Methods for Creating Timetables on Database Ben Paechter [12]

Large		Medium5		Medium4		Medium3		Medium2		Medium1		Small5		Small4		Small3		Small2		Small1		Datasets Algorithms
M	B	M	B	M	B	M	B	M	B	M	B	M	B	M	B	M	B	M	B	M	B	Best/Med
-	100%In	152.6	151	183.6	181	267.8	265	262.8	161	245	242	0	0	0	0	0	0	0	0	0	0	A_5
-	-	-	1068	-	171	-	348	-	359	-	419	-	372	-	3	-	3	-	7	-	6	A_6
-	529	-	135	-	165	-	246	-	147	-	221	-	0	-	0	-	0	-	0	-	0	A_7
-	876	-	141	-	112	-	189	-	130	-	140	-	0	-	6	-	6	-	4	-	3	A_9
-	-	203	200	155	142	238.5	235	185	180	229.5	227	0	0	0	0	0	0	0	0	0	0	A_{25}
-	-	-	292	-	247	-	357	-	313	-	317	-	0	-	0	-	0	-	0	-	0	A_{26}
-	80%In	-	303	-	169	-	267	-	173	-	146	-	0	-	1	-	0	-	2	-	1	A_{27}
100%In	-	100%In	-	177.5	-	77.5%In	-	202.5	-	199	-	5	-	7	-	8	-	11	-	8	-	A_{28}
-	-	-	232	-	247	-	249	-	188	-	280	-	0	-	0	-	0	-	3	-	0	A_{29}
851.5	-	219.5	-	164.5	-	248	-	184	-	195	-	0	-	1	-	1	-	3	-	1	-	A_{30}
-	1138	-	112	-	285	-	249	-	325	-	243	-	7	-	17	-	7	-	9	-	10	A_{31}
622.5	615	119.5	116	101	98	124	122	69.5	92	143	139	0	0	0	0	0	0	0	0	0	0	A_{32}
-	1027	-	276	-	321	-	251	-	258	-	254	-	4	-	0	-	2	-	4	-	2	A_{33}
-	100%In	-	232	-	247	-	249	-	188	-	280	-	0	-	0	-	0	-	3	-	0	A_{34}

6. Conclusion

- Among the methods based on PSO, HPSO and SPSO were more efficient than PSO. Adding the local search to each of PSO and SPSO, their efficiencies can increase.
- HHSA is based on the harmony search. It was a high efficiency compared with 26 other methods.
- Comparing the methods based on ant colony, the efficiencies of BWAS and BWACS were higher than ACO. It can be stated that BWACS was more appropriate for large problems, and BWAS was better for small problems.
- Hybrid methods of ACS-TS and ACS-SA were proposed to improve the efficiency of ant colony optimization. The evaluation results indicated that hybrid methods were more efficient than ant colony.
- Comparing ant colony optimization with the simulated annealing algorithm, it can be stated that ant colony optimization indicated a higher quality and efficiency. Moreover, the evaluation results indicated that if the local search is added to the improved method, its efficiency will considerably increase.
- In general, it can be stated that hybrid methods were more efficient than not hybrid methods.
- Comparing different methods with each other, it can be stated that the memetic algorithm and PSO were more efficient than the genetic algorithm. Honey bee optimization algorithm was better than the genetic, memetic, and ant colony optimization algorithms. The harmony search algorithm was more efficient than the majority of methods such as ant colony, fuzzy, the genetic and memetic algorithms.

7. Future Works

So far, many methods have been proposed, discussed, and evaluated by different researchers for the educational timetabling problem, although there is still work to be done in the future. Some of the future works to create educational timetables which can be discussed are as follows:

- Using the forest algorithm to solve the educational timetabling problems of universities
- Designing parallel algorithms to create the educational timetables with the available methods to increase their efficiencies

References

[1]. Der-Fang, S.; A hybrid sectionicle swarm optimization for a university course scheduling problem with flexible preferences; Expert Syst. Appl. Volume 38, Pages 235-248, (January 2011).

[2]. Qaurooni, D., Akbarzadeh, M.R.; Course timetabling using evolutionary operators; Applied Soft Computing; Volume 13, Issue 5, Pages 2504-2514, (2013).

[3]. Asmuni, H., Burke, E.K., Garibaldi, J.M.; Fuzzy Multiple Heuristic Ordering for Course Timetabling; In: Proceedings of 5th UK Workshop on Computational Intelligence (UKCI ’05), Pages 302–309, (2005).

[4]. Thepphakorn, T., Hicks, Ch., Pongcharoen, P.; An Ant Colony Based Timetabling Tool; International Journal of Production Economics, Volume 149, Pages 131–144, (March 2014).

[5]. Azmi Al-Betar, M., Tajudin Khader, A.; A harmony search algorithm for university course timetabling. Annals of Operations Research, Volume 194, Issue 1, Pages 3-31, (April 2012).

[6]. Chen, R.-M.; Shih, H.-F. Solving University Course Timetabling Problems Using Constriction Sectionicle Swarm

Optimization with Local Search. Algorithms 2013, 6, 227-244.
[7]. Adrianto, D.; Comparision Using Sectionicle Swarm Optimization and Genetic Algorithm for Timetable Scheduling; Journal of Computer Science 10 (2): 341-346, 2014.
[8]. Zadeh L. A., Fuzzy Logic: Computing with Words, IEEE Transactions on Fuzzy Systems (TFS), vol. 4, no. 2, pp. 103-111, May 1996.
[9]. Mamdani E. H., Advances in the Linguistic Synthesis of Fuzzy Controllers, International Journal of Man-machine Studies (IJMMS), vol. 7, no. 1, pp. 1-13, 1976.
[10]. Azmi Al-Betar, M., Tajudin Khader, A.; University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm; Applications and Reviews, VOL. 42, no. 5, September 2012.
[11]. Ayob, M., Jaradat, G.; Hybrid Ant Colony Systems For Course Timetabling Problems. IEEE 2nd Conference on Data Mining and Optimization 27-28 October 2009, Selangor, Malaysia, 120-126, 2009.
[12]. Babaei, H., Karimpour, J., Hadidi, A., A Survey of Approaches for University Course Timetabling Problem, Computers & Industrial Engineering 2014;
[13]. Badoni, R. P., Gupta, D.K., Mishra, P.; A new hybrid algorithm for university course timetabling problem using events based on groupings of students; Computers & Industrial Engineering vol.78, pp. 12–25, 2014.
[14]. Ozcan, E., Alkan, A.; A memetic algorithm for solving a timetabling problem: An incremental strategy; Proc. of the 3rd Multidisciplinary Int. Conf. On Scheduling: Theory and Applications; pp. 394-401, At Paris, France, 2014;
[15]. Mayer, A., Nothegger, C., Chwatal, A., Raidl, G.; Solving the Post Enrolment Course Timetabling. Problem by Ant Colony Optimization. Annals of Operations Research; Vol. 194 Issue 1, p 325-339, Apr2012.
[16]. Chmait, N., Challita, K.; Using Simulated Annealing and Ant-Colony Optimization Algorithms to Solve the Scheduling Problem; Computer Science and Information Technology 1(3): 208-224, 2013;
[17]. Sabar, N. R., Masri, A., Kendall, G., Qu, R.; A honey-bee mating optimization algorithm for educational timetabling problems; European Journal of Operational Research 216 (2012) 533–543.
[18]. M. Alzaqebah a,n, S. Abdullah; Hybrid bee colony optimization for examination timetabling problems; Computers & Operations Research 54 (2015) 142–154.
[19]. Schaerf, A.; A Survey of Automated Timetabling; Artificial intelligence publishers; 1/02/1999; 16:48; no v.; p.3.
[20]. Kristiansena, S., Stidsena , T. R.; A Comprehensive Study of Educational Timetabling a survey; Desectionment of Management Engineering, (Nonember 2013)
[21]. Pillay, N., A survey of school timetabling research; Springer (2013)
[22]. Fong, C.W., Asmuni, H.b., McCollum, B., McMullan, P., Omatu, S.; A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems, Information Sciences (2014), doi: http://dx.doi.org/10.1016/j.ins.2014.05.039
[23]. http://tabu.diegm.uniud.it/ctt/index.php?page=rankings&see=best#table
[24]. [24]. Afsari, F., Eftekhari, M., Zahedi, M.; Planning Timetabling University Cource Using Ant_fuzzy; 14th Annual Conference of computer society of iran, Tehran, Amirkabir University (1387)
[25]. [25]. Doulaty, M., Feizi Derakhshi, M. R, Abdi. M.; Timetabling: A State-of-the-Art Evolutionary Approach; International Journal of Machine Learning and Computing, Vol. 3, No. 3, (June 2013)