♦ Recepción/ 27 junio 2019
♦ Aceptación/ 25 agosto 2019

Solving Examination Timetabling Problem Using Crow Swarm Optimization Algorithm

Resolución de problemas de examen de horarios utilizando el algoritmo de optimización Crow Swarm

Amal Turki Mahdi,
College of Computer Science and IT
University of Anbar, Ramadi, Iraq
amoola.turky94@gmail.com

Belal Al-Khateeb,
College of Computer Science and IT
University of Anbar, Ramadi, Iraq
belal@computer-college.org

ABSTRACT/ Examination Timetabling Problem (ETP) is a part of the Timetabling problem, which belongs to the set of combinatorial optimization problems as well as is of great significance for each University. It is addressed by heuristic and/or artificial intelligent methods. Generally, ETP involves entering a collection of exams in a specific period of time according to a set of diverse constraints that must be satisfied. Because of their wide applications, the researches are still underway on ETP. This paper presents Crow Swarm Optimization (CSO) technique which is a new intelligent way that has been proposed for the examination timetabling problems, which inspired by the behavior of crow swarms in nature. This type of crows found in North America so it is called American Crow. The CSO algorithm simulates the cooperative behavior of crows during the search for food. In nature, American Crow is divided into groups to search for food. With these strategies, it achieves the diversity in solutions and the individuals in the new algorithm explore and exploit the search space more efficiently. The research deals with the uncapacitated examination timetabling problem. The proposed algorithm is verified in 7 instances of examination timetabling. According to the results of the experiments, the proposed algorithm can provide a promising set of solutions for each examination timetabling instance by comparing our proposed algorithm with Bee Colony Optimization (BCO) and BCO with tournament selection strategy. In addition, a comprehensive comparison was conducted with the best known results on uncapacitated examination timetabling datasets, which across all the problem instances displays that the proposed approach is competitive and also works well.
Keywords: Examination Timetabling Problem, Crow Swarm Optimization, Constraints Satisfaction, Search Methodologies, Toronto Benchmark Datasets.
RESUMEN/ El examen de problemas de horarios (ETP) forma parte del problema de horarios, que pertenece al conjunto de problemas de optimización combinatoria y es de gran importancia para cada universidad. Se aborda mediante métodos inteligentes heurísticos y / o artificiales. En general, ETP implica ingresar a una colección de exámenes en un período de tiempo específico de acuerdo con un conjunto de restricciones diversas que deben cumplirse. Debido a sus amplias aplicaciones, las investigaciones aún están en curso sobre ETP. Este artículo presenta la técnica de optimización de enjambres de cuervos (CSO), que es una nueva forma inteligente que se ha propuesto para el examen de los problemas de cronometraje, que se inspiró en el comportamiento de los enjambres de cuervos en la naturaleza. Este tipo de cuervos se encuentra en América del Norte, por lo que se llama American Crow. El algoritmo CSO simula el comportamiento cooperativo de los cuervos durante la búsqueda de alimentos. En la naturaleza, American Crow se divide en grupos para buscar comida. Con estas estrategias, logra la diversidad de soluciones y los individuos en el nuevo algoritmo exploran y explotan el espacio de búsqueda de manera más eficiente. La investigación aborda el problema de horarios de exámenes no capacitados. El algoritmo propuesto se verifica en 7 casos de examen de horarios. De acuerdo con los resultados de los experimentos, el algoritmo propuesto puede proporcionar un conjunto prometedor de soluciones para cada instancia de cronometraje de exámenes al comparar nuestro algoritmo propuesto con la Optimización de colonias de abejas (BCO) y BCO con la estrategia de selección de torneo. Además, se realizó una comparación exhaustiva con los mejores resultados conocidos en conjuntos de datos de examen de horarios no capacitados, que en todas las instancias del problema muestran que el enfoque propuesto es competitivo y también funciona bien.
Palabras clave: problema de examen de horarios, optimización de enjambres de cuervos, satisfacción de restricciones, metodologías de búsqueda, conjuntos de datos de referencia de Toronto.

1. Introduction

The way of obtaining the best solution from the set of available solutions, with consideration all constraints needed for a specific problem, is called the optimization process [1]. In optimization problems, especially in the single objective category, which consists of obtaining the maximum or the minimum value of an objective function that computes the quality of each solution. The variables of optimization problems can be either discrete or continuous. Usually, the discrete optimization problems also named as combinatorial optimization problems; the applied ETP in this paper is within this class of problems [1].

Although there are many methods and algorithmic strategies are adapted to solve hard and complicated combinatorial optimization problems, none of them could efficiently solve all the classes of optimization problems, as Wolpert and Macready [2] proved. Some algorithmic strategies that apply to these problems are mathematical programming [3], artificial intelligence [4] and meta-heuristic techniques [5]. The class of swarm intelligent (SI) algorithms that is used in this paper, is the collective behavior of decentralized, self-organized systems, natural or artificial and is usually belongs to computational intelligence.

Many types of research have been executed to solve the Examination Timetabling Problem (ETP) of the universities to have a created automated system. In practical, the ETP is combinatorial problems that comprise assigning a set of examinations into an identified set of time slots while achieving a set of hard constraints that cannot be broken that and a set of soft constraints that must be minimized as possible [6]. As known that ETP fall under the NP-hard problems [7], [8]. Moreover, it a dynamic and perturbed problem. Hence, various metaheuristic methods have been developed for ETP that can be classified into two main classes, single-based methods (e.g. variable neighborhood-based search, tabu search, deluge and great simulated annealing) and population-based methods (e.g. ant colony optimization, genetic algorithms and memetic algorithms) [9]. Single-based interest methods have got by many various researchers because of the efficacy of these methods to exploit the search space in a short have time. However, these methods have limitations such a weak exploration so it is easy to get stuck in a local optimum [9].

Population-based methods are presented to solve complex problems such as examination timetabling. The main purpose of population-based is to concentrate on iterative improvement for a set of solutions [10]. Population-based methods are classified as either Evolutionary Algorithms (EA) for instance Genetic Algorithm (GA)[11] , Genetic Programming (GP) [12] and Differential Evolution (DE) [13] or Physics-Based (PB) like Gravitational Search Algorithm (GSA) [14] and Curved Space Optimization (CSO) [15] or Swarm Intelligence (SI)-based algorithms like Salp Swarm Algorithm (SSA)[16] and Gray Wolf Optimizer (GWO) [17]. SI features motivated many researchers to apply such behavior for optimization problems. SIs, including an Ant Colony Optimization (ACO) [18], Fish Swarm Optimization (FSO) algorithm [19], Artificial Bee Colony (ABC) algorithm [20] and Meerkat Swarm Optimization (MSO) [21] have been widely used to solve ETP in the literature. The paper is organized as follows:

Section 2 explains ETP and Previous studies that are close. **Section 3** describes the proposed CSO and the way that it applies to solve ETP. **Section 4** displays the experimental result of the study after performing the proposed method for solving ETP and **section 5** comprises a conclusion and consideration for future research.

2. Examination Timetabling Problem

Examination timetable is one of the most significant administrative activities that arise in various academic institutions. Usually, it is hard to handle, complex and time-consuming. In practical, the most popular scenario in the scheduling is to avoid overlapping of examinations for every course with common enrolled students. ETP can be clarified as allocating a set of exams $E = \{e_1, e_2, e_3, ..., e_e\}$ into a few specified timeslots (periods) $T = \{t_1, t_2, t_3..., t_t\}$ and rooms of a particular size in each timeslot $C = \{C_1, C_2, C_3..., C_t\}$, directed to a collection of constraints. Difficulty and complexities cases that usually arise in timetabling problems from the truth that there is a large diversity of constraints require to be achieved in many institutions and often some of these constraints oppose each other [9]. Generally, the constraints are categorized into two categories: hard and soft constraints that are illustrated below:

- **Hard Constraints:** under any circumstances, cannot break the hard constraints. For instance, interrelated examinations that contain shared resources such as students can't be scheduled at the same time, i.e. if S_{ij} refers to students number that enrolled in both exams i and j; and $t_i \in T$ is the timeslot to which exam i is assigned, then:

$$t_i \neq t_j \; \forall \, i,j \in E, i \neq j \text{ and } S_{ij} > 0$$

Usually, a timetable that satisfies all hard constraints is said to be feasible.

- **Soft Constraints:** Described as desirable but are not essential. They vary from one to another institution in terms of both the importance and their varieties. However, soft constraints sometimes oppose each other so It's impossible to get workable solutions that achieve all soft constraints. The popular soft constraint in the examination timetabling can be presented as separate the conflict exams as possible in any examination period, hence the students will have sufficient time to review between exams. The exam timetables quality measured by checking the soft constraints which may break during the generation of solutions [22].

Two types of the ETP problem exists, the capacitated and the uncapacitated (also called Toronto datasets). If the room size is taken into account, then this refers to the capacitated type. Otherwise is the uncapacitated type [22]. In this work, the Toronto datasets are adopted. In this datasets, only one hard constraint is contained which is about the fact that a student must not sit in over one examination at the same time. The main idea in soft constraint is spreading the examinations as much as possible into the whole periods of examinations to facilitate the preparation of the students.

Many studies have been directed to solve ETP with the use of various strategies or methods to provide an efficient, reliable and acceptable schedule for the exams. The following presents some related work for solving ETP.

Eley (2007) proposed ant algorithms for solving the examination problem. The author studied two strategies which are Max-Min and ANTCOL. The hybridization process between a simple ant system and hill climber was made in both strategies. The construction of initial solutions based on the inverse of degree metaheuristic. The author noticed that the sim-ANTCOL outperformed the hybrid hill Max–Min Ant system with hill climber. Also noticed that the sitting of parameters in the system of ant algorithm affects the performance of strategies. This is useful in finding new solutions premature and capable to avoid the convergence. As reported, the drawbacks are slow, requires parameters and not effective as an improvement algorithm [23].

E. K. Burke et al. (2010) offered an alternative of variable neighborhood search for solving uncapacitated exam timetabling problems. Indirectly, the genetic algorithm applied to effectively choose a subset of neighborhoods. Better results were collected on some Carter benchmark problem instances [24].

N. Pillay et.al (2010) suggested a two-phase knowing genetic algorithm for examination timetabling problems. In the first phase, the concentration on generating workable timetables that achieve all hard constraints, whilst in the second phase attempts to reduce violations of the soft constraint. In both phases, the genetic algorithm applied in order to generate and advance the timetable. Similar results on the Carter benchmark datasets have been obtained across other approaches. Though, the authors did not discuss the purpose behind applying a GA in the construction and enhancement phases, so this led to increases the computational time and the number of parameters that require to be attuned. All the results that gained are worse than the best result known [25].

M. Alzaqebah et.al (2014) presents a hybrid Bee Colony Optimization (BCO) approach to solve the examination timing problems. the algorithm performs two activities, the forward and backward pass. The bees investigate the search area in the forward and gives the information to other bees in the colony in the backward pass. The researchers noticed that each bee determines to wither explores for a food source as a recruiter or selects a recruiter bee to follow as a follower. In the forward pass, the algorithm is carried beside other local searches, such as Simulated Annealing and the Late Acceptance Hill Climbing algorithms. The researchers proposed three strategies are the rank, tournament and disruptive selection strategies. These strategies for the follower bees in order to choose a recruiter to keep population variety in the backward pass. Also, they proposed a mechanism to choose a neighborhood structure to improve the neighborhood search called a self-adaptive mechanism. However, the proposed algorithm is verified against some methodologies with consideration examination timetabling problems uncapacitated and capacitated datasets. The experimental results produce better results on uncapacitated and comparable results on capacitated datasets [26].

3. Problem Description

In the sections, the Toronto datasets, the proposed CSO and solve ETP by CSO algorithm are described.

3. 1 Uncapacitated Toronto Dataset

The common examination benchmark dataset that widely used was introduced by [27]. This type of dataset also known as the Toronto benchmark dataset. Basic version of these dataset is available at http://www.asap.cs.nott.ac.uk/resources/data.shtml. This dataset considered as an uncapacitated dataset for examination timetabling, because of the assumption of an unlimited number of seats during exam assignment. The Toronto dataset includes 13 problem instances. In Table 1, the characteristics and details information of the dataset.

Table 1: The Characteristics of Toronto Datasets [28].

Problem Name	Time-Slots	Exams	Students	Conflict Density
Car-f-91	35	682	16 925	0.13
Car-s-92	32	543	18 419	0.14
Ears-f-83	24	190	1 125	0.27
Hec-s-92	18	81	2 823	0.42
Kfu-s-93	20	461	5 349	0.06
Lse-f-91	18	381	2 726	0.06
Pure-s-93	42	2419	30 029	0.03
Rye-s-92	23	486	11 483	0.08
Sta-f-83	13	139	611	0.14
Tre-s-92	23	261	4 360	0.18
Uta-s-92	35	622	21 266	0.13
Ute-s-92	10	184	2 749	0.08
Yor-f-83	21	181	941	0.29

The main objective of the Toronto problems is to generate a feasible timetable, in which no student is required to sit more than one examination at the same time. In order to obtain a high quality feasible timetable then the soft constraints must be achieved as much as possible. Therefore, it is essential to separate student's examinations as far as possible during the construction of the timetable, in order to give a wide spread for the student in the timetable. In Toronto datasets, there is one soft constraint only that prevents each student to get more than exam in close timeslots or periods. The soft constraint expressed by the formula f_c, defined in (1). The formulated cost function is to minimize [29]:

$$f_c = \frac{\sum_{i=1}^{E-1}\sum_{j=i+1}^{E} S_{ij} * prox(i,j)}{M} \qquad (1)$$

Where

$$prox(i,j) = \begin{cases} 2^{5-|t_i-t_j|}, & if\ 1 \le |t_i - t_j| \le 5 \\ 0, & othrwise \end{cases}$$

Subject to:

$$\sum_{i=1}^{E-1}\sum_{j=i+1}^{E} S_{ij} * \delta_{ti,tj} = 0$$

$$\delta_{ti,tj} = \begin{cases} 0, & t_i \neq t_j \\ 1, & t_i = t_j \end{cases} \qquad (2)$$

The uncapacitated ETP terms are following defined:

- $E = \{e_1, e_2, \ldots, e_e\}$ is the set of exams.
- $T = \{t_1, t_2, \ldots, t_k\}$ is the set of timeslots (or periods).
- S_{ij} = denotes to the students that attending to both exams i and j.
- M the whole number of students.
- t_k ($t_k \in P$) indicates the allocated timeslot for exam e_k ($e_k \in E$).

Equation (1) provides the penalty acquired by allocating the exams ei and ej into the timeslots ti and tj, respectively. The weighting factor for the penalty is 16, 8, 4, 2 and 1, for the exams one, two, three, four and five timeslots respectively. The weighting factor set to zero value if the gap between the exams over than five timeslots apart. Equation (2) denotes to a hard constraint, which requires no clash between the scheduled exams in the same period[33][34][35].

The high-quality timetable achieved if the penalty value equal to zero. This means each student has a gap between their exams, which is at least a five slots between one and the next exam session. In general, none of the values got in the examination timetable especially Toronto dataset has a zero penalty. This denotes to the difficulty of the real-life problem, hence Intensify efforts in order to obtain a shorter period for the examinations.

3.2 Crow Swarm Optimization Algorithm

Crow Swarm Optimization (CSO) is a new intelligent method inspired by the behavior of crow swarms in nature. This type of crows found in North America so it is called American Crow. The CSO algorithm simulates the cooperative behavior of crows during the search for food as shown in algorithm 1. To model such interactions, every cluster of crows' area unit needed to maneuver over the search area. The crows when being divided into teams who begin to look for places of food at long distances area and not among the scope of traditional vision. Assume the crow's algorithmic program determines the simplest cluster you get when choosing the food space and additionally deciding the totals that didn't get sensible food on this trip. Within the next journey of food search, teams with dangerous food can eat sensible food. Reckoning on the characteristics of the animal, like speed, angle for departure and placement. With these strategies, it achieves the diversity in solutions and the individuals in the new algorithm explore and exploit the search space more efficiently.

Algorithm 1: Crow Swarm Optimization

INPUT: Crows population Pi (i=1…n), speed S (S ϵ [-6, 6]), angle ϴ (ϴ ϵ [45,135]).

Distribute the Crows randomly into groups (G), where each group will have different number (must be greater than 3) of Crows.
Calculate the fitness of all crows in each group.
For each group, select the best value of fitness and store it in CurrentBest.
t=1.

While (t≤ maximum number of iterations)

Update the position of all Crows in each group
Update the fitness of all Crows
For each group, select the best value of fitness and store it in NewBest.
Update CurrentBest:
if NewBest is better than CurrentBest then CuurentBest=NewBest.
Update the angle of crow
t=t+1.

End while.

OUTPUT: Best crow in all groups.

3.3 Solving ETP by CSO Algorithm

In this section, the proposed CSO is applied to the ETP.

3.3.1 Dataset Representation

In this study, to alleviates the formats variety as well as dealing with the initial problem of the dataset by implementing an algorithm or function in order to change the format of a dataset into a standard format, which will be employed as input data to the pre-processing phase. Some matrices or data samples that must be identified and employed in order to maintain the original and processed data inside the system. The major sample of data is the StudentExam array, which expresses the correlation between a student and every exam enrolled by that student as shown in figure 1.

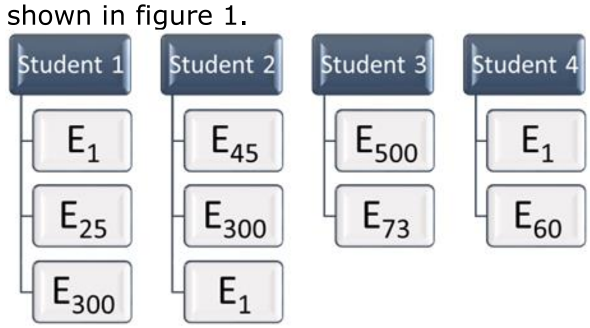

Figure 1: Representation of StudentExam Matrix Example.

Based on the StudentExam matrix, it is possible to create another matrix called

ExamStudent that describes the relationship between each exam and the students enrolled on it (see figure 2). This matrix will facilitate the implementation when using the algorithm.

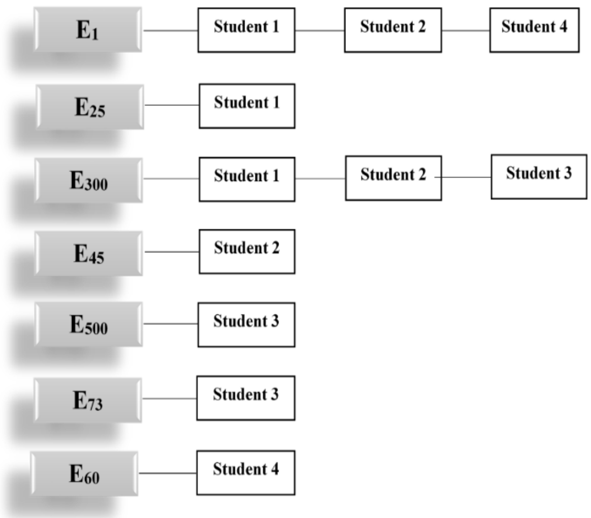

Figure 2: ExamStudents Matrix Generated Based on the StudentExam Matrix.

3.3.1 Solutions Initialization

The adopted solution representation for the Toronto dataset is illustrated in figure 3. Where assigning exams to random periods within the initialization phase.

t_1	t_2	t_n
e_6	e_1	e_8
e_{14}	e_{11}	e_{12}
e_{10}	e_4	e_{15}
e_3	-	e_{17}
e_{16}	-	-

Figure 3: Solution Representation for the Toronto Dataset.

Each solution (exam timetable) clarified as an array of timeslots and each timeslot comprises a set of the exams that are scheduled in that timeslot. For instance, in figure 3, the exams that assigned to timeslot t_1 are e_6, e_{14}, e_{10}, e_3 and e_{16}, while the exams that assigned to timeslot t_2 are e_1, e_{11} and e_4. The exams that assigned to a specific timeslot must achieve the hard constraint. After achieving hard constraint, the feasible solution is obtained. At first, the speed and angle (parameter required by CSO) generated randomly for each solution. The penalty (fitness) calculated for each solution then divide the solutions (exam tables) into two subgroups with equal size. select the best solution for each subgroup as BestCurrentTable, which is the solution that has the minimum penalty amongst them.

3.3.2 Solution Processing

After the initialization method is completed, the CSO algorithm starts to work by updating the period (timeslot) for each exam in exam table. In the light of calculating penalty for each exam table, the best exam table is determined for each subgroup as New Best Table. The updating of Best Curent Table depends entirely on New Best Table, therefore, if the penalty of New Best Table better (minimum) than BestCurentTable penalty then Best Curent Table take penalty value of New Best Table. Otherwise, the situation remains as it is. The angle which is a parameter required for CSO algorithm is updated before ending of iteration. This process will continue until reaching the stop condition then the better Best Curent Table returned as the best exam table. Algorithm 2 shows the CSO for ETP algorithm.

Algorithm 2: Crow Swarm Optimization for Examination Timetabling.

Initialization

INPUT: Get The Dataset Information.

- Initialize N ExamTable solutions.
- Distribute ExamTable randomly into subgroups (G)
- Calculate Penalty of each ExamTable.
- Select the best solution in each subgroup as BestCurrentTable.
- t=1.
- Max number of iterations.

Solution

While (t < Max number of iterations)

- Update Period of all ExamTable in each subgroups
- Calculate Penalty of each ExamTable.
- Select the best solution in each subgroup as NewBestTable.

- Update BestCurrentTable:
  - if NewBestTable is better than BestCurrentTable then BestCurrentTable = NewBestTable.
- Update the angle of ExamTable
- t=t+1.

End while.
OUTPUT: BestCurrentTable in all subgroups.

4. Results and Discussion

In this work, we used data from the dataset called "Toronto" in order to solve ETP; the data is specifically selected from the categories, which are (Ear-f-83, Hec-s-92, Kfu-s-93, Les-f-91, Sta-f-83, Ute-s-92 and Yor-f-83). Table 2 shows the results of these datasets according to the following settings: population size 50 and 10 runs with 1000 iterations for each run.

Table 2: Results of Ear-f-83, Hec-s-92, Kfu-s-93, Les-s-91, Sta-f-83, Ute-s-92 and Yor-f-83 by CSO Algorithm.

Dataset	**Runs**				
Ear-f-83	40.1831	39.8809	40.1067	39.5502	39.6578
	39.8569	39.4453	39.4791	39.2738	**38.9591**
Hec-s-92	12.1773	11.6733	11.3656	11.5054	11.3291
	12.1961	**11.2682**	11.3376	12.2774	11.4938
Kfu-s-93	16.609	16.716	16.498	16.318	17.231
	16.606	16.536	16.895	16.233	**16.172**
Les-f-91	12.3955	**12.2058**	13.9376	13.4266	12.803
	12.3376	12.2774	13.7201	13.7898	12.7381
Sta-f-83	159.9198	160.8363	160.2144	159.6628	159.2799
	159.7201	159.8036	159.3656	**159.2526**	160.2488
Ute-s-92	28.2182	28.3289	28.3379	28.1332	28.5743
	29.1428	27.4136	**27.1356**	29.1281	27.5358
Yor-f-83	39.1081	39.0613	39.076	39.2289	39.3412
	38.1237	38.2215	39.3313	38.5677	38.3794

Table 2 shows the executions for each dataset at which the best value is in bold. To measure the efficiency and strength of CSO algorithm. First, we compared the results of this algorithm with BCO as well as BCO with tournament selection strategy (TBCO) in [17]. The experimental results shown in table 3 and table 4. Also, CSO is compared with some other selected algorithms as shown in table 5.

Table 3: Comparison between CSO and BCO.

Dataset	**CSO**				**BCO**			
	Min	***Max***	***Ave.***	***Std.***	***Min***	***Max***	***Ave.***	***Std.***
Ear-f-83	38.95	40.18	39.63	0.37	**38.32**	39.92	39.12	0.51
Hec-s-92	**11.26**	12.27	11.66	0.4	11.44	11.99	11.67	0.19
Kfu-s-93	16.17	17.23	16.58	0.31	**16.01**	16.96	**16.48**	0.26
Lse-f-91	**12.2**	13.93	12.96	0.68	13.27	13.84	13.57	0.18
Sta-f-83	159.25	160.83	159.83	0.49	**157.81**	158.24	158.04	0.13
Yor-f-83	**38.12**	39.34	38.84	0.47	38.17	39.53	38.95	0.38
Ute-s-92	**27.13**	29.14	28.19	0.67	27.16	29.07	28.25	0.49

According to the results of the above table, the CSO algorithm gets the best minimum value in four datasets (Hec-s-92, Les-f-91, Yor-f-83 and Ute-s-92) when compared to BCO. However, the results of the remaining datasets were competitive compared to BCO.

Table 4: Comparison between CSO and TBCO.

Dataset	CSO				TBCO			
	Min	*Max*	*Ave.*	*Std.*	*Min*	*Max*	*Ave.*	*Std.*
Ear-f-83	38.95	40.18	39.63	0.37	**38.35**	39.86	39.15	0.54
Hec-s-92	**11.26**	12.27	11.66	0.4	11.42	11.98	11.7	0.18
Kfu-s-93	**16.17**	17.23	16.58	0.31	16.54	17.17	16.79	0.18
Lse-f-91	**12.2**	13.93	12.96	0.68	13.29	14.02	13.65	0.2
Sta-f-83	159.25	160.83	159.83	0.49	**157.85**	158.3	158.01	0.14
Yor-f-83	**38.12**	39.34	38.84	0.47	38.18	39.65	39.02	0.53
Ute-s-92	27.13	29.14	28.19	0.67	**27.07**	28.87	28.19	0.63

Table 4 presents the results of the CSO algorithm compared to TBCO algorithm, where CSO gets the best value in the four datasets (Hec-s-92, Kfu-s-93, Les-f-91 and Yor-f-83) when compared to TBCO algorithm. The results in the remaining datasets were competitive compared to TBCO. It is worth to mention here that the time taken to implement these datasets is varied and it may take up to six hours based on the size of the problem instance. In a clearer sense, this is because of the number of students in the dataset, where the larger number of students participating in the exams leads to greater time spent in implementing the data set, so it takes time to achieve a hard constraint. Anyway, this runtime is acceptable in university timetabling problems because the timetables are usually produced months before the actual schedule is required.

Table 5: Performance Comparison of CSO with The Previously Used Methods According to The Best Penalty Got.

Approach, [Source]	Hec-s-92	Kfu-s-93	Lse-f-91	Sta-f-83	Ute-s-92	Yor-f-83
SD, [30]	12.7	15.9	12.9	165.7	31.5	44.8
LWD, [30]	15.8	22.1	13.1	161.5	26.7	41.7
LE, [30]	15.9	20.8	**10.5**	161.5	**25.8**	45.1
FZLO, [31]	11.7	15.8	12.09	160.4	27.7	40.5
GCCHH, [32]	11.9	**15.3**	12.33	160.12	32.67	40.53
CSO	**11.2**	16.1	12.2	**159.2**	27.1	**38.12**

Note: Saturation Degree (SD), Largest Degree (LD) and Largest Weighted Degree (LWD) are algorithmic strategies proposed by Carter et.al (1996). FZLO: Fuzzy multiple graph coloring ordering criteria proposed by Asmuni et al. (2005). GCCHH: A Graph Coloring Constructive Hyper-Heuristic proposed by N. R. Sabar et.al. (2012). The results in table 5 show that CSO obtained comparable results for all tested instances across all approaches. It obtains the best result solutions on three (Hec-s-92, Sta-f-83 and Yor-f-83) instances. This indicates the ability of this algorithm to solve problems of this type.

5. Conclusions and Future work

In recent years, the optimization algorithms especially the algorithms that simulate the behaviors of nature or specific phenomena in the optimization computation are utilized in various areas to get the best solutions for complex problems across heuristic approaches. One of the most important and complex optimization problems in most colleges and universities are the timetabling problem. This paper presents the CSO method that applies to solve real-world ETP. This method handles the time-consuming that spent on preparing examination timetable manually, which usually requires several days of work with the final solution still sometimes unsatisfactory as it is an NP-hard problem. according to the results in table 2 thru 4, the obtained results demonstrate that CSO is a very good algorithm for solving ETP like problems. In addition, the performance of CSO is compared against popular previous approaches in table 5. The results showed a superior performance for the most tested instances. Several directions can be recommended for future studies such apply CSO to similar combinatorial optimization problems, such as course timetabling, vehicle routing and airline timetable problems to ensure the generality of our proposed algorithm across different applications.

References

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge University Press, 2004.
[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.
[3] G. B. Dantzig, *Linear Programming and Extensions*, no. August. Princeton University Press: Princeton, 1963.
[4] J. Mccarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "Dartmouth summer research project on artificial intelligence," vol. 27, no. 4, 2006.
[5] J. K. Mandal, S. Mukhopadhyay, and T. Pal, *Handbook of Research on Natural Computing for Optimization Problems*, vol. i. 2016.
[6] Mohammed, M.A., Ghani, M.K.A., Mostafa, S.A. and Ibrahim, D.A., 2017. Using Scatter Search Algorithm in Implementing Examination Timetabling Problem. *Journal of Engineering and Applied Sciences*, *12*, pp.4792-4800.
[7] A. Schaerf, "A Survey of Automated Timetabling," no. Gotlieb 1963, pp. 87–88, 1999.
[8] T. B. Cooper and J. H. Kingston, "The complexity of timetable construction problems," in *International Conference on the Practice and Theory of Automated Timetabling*, 1995, pp. 281–295.
[9] R. Q. E. K. Burke, B. M. L. T. G. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," pp. 55–89, 2009.
[10] GHAIDA MUTTASHAR ABDULSAHIB and OSAMAH IBRAHIM KHALAF, 2018. AN IMPROVED ALGORITHM TO FIRE DETECTION IN FOREST BY USING WIRELESS SENSOR NETWORKS.International Journal of Civil Engineering & Technology (IJCIET) - Scopus Indexed.Volume:9,Issue:11,Pages:369-377.
[11]J. H. Holland, "Genetic algorithms," *Sci. Am.*, vol. 267, no. 1, pp. 66–73, 1992.
[12]J. Koza, "Genetic programming," *Encycl. Comput. Sci. Technol.*, no. 1989, pp. 1–26, 1997.
[13]R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[14]E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," Elsevier, 2009.
[15]F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, "Curved space optimization: a random search based on general relativity theory," *arXiv Prepr. arXiv1208.2214*, 2012.
[16]S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, 2017.
[17] Osamah Ibrahem Khalaf, Ghaidaa Muttasher et al., "Improving video Transmission Over Heterogeneous Network by Using ARQ and FEC Error Correction Algorithm", vol. 30, no.8, pp.24-27, Nov 2015
[18]K. A. Dowsland and J. M. Thompson, "Ant colony optimization for the examination scheduling problem," *J. Oper. Res. Soc.*, vol. 56, no. 4, pp. 426–438, 2005.
[19]H. Turabieh and S. Abdullah, "A hybrid fish swarm optimisation algorithm for solving examination timetabling problems," in *International Conference on Learning and Intelligent Optimization*, 2011, pp. 539–551.
[20]M. Alzaqebah and S. Abdullah, "Hybrid artificial bee colony search algorithm based on disruptive selection for examination timetabling problems," in *International Conference on Combinatorial Optimization and Applications*, 2011, pp. 31–45.
[21]A.-K. Belal and T. Amal, "Meerkat Swarm Optimization Algorithm for Real World University Examination Timetabling Problem," *Adv. Res. Dyn. Control Syst.*, vol. 11, pp. 2103–2112, 2018.
[22]S. K. N. Abdul Rahim, "Transformation of the university examination timetabling problem space through data pre-processing," Nottingham, 2015.
[23]M. Eley, "Ant algorithms for the exam timetabling problem," in *International Conference on the Practice and Theory of Automated Timetabling*, 2006, pp. 364–382.
[24]E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu, "Hybrid variable neighbourhood approaches to university exam timetabling," *Eur. J. Oper. Res.*, vol. 206, no. 1, pp. 46–53, 2010.

[25]N. Pillay and W. Banzhaf, "An informed genetic algorithm for the examination timetabling problem," vol. 10, pp. 457–467, 2010.
[26]M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Comput. Oper. Res.*, 2014.
[27]M. W. C. Laporte, "Recent developments in practical examination timetabling," *Springer, Berlin, Heidelb.*, 1996.
[28]V. Kolonias, G. Goulas, C. Gogos, P. Alefragis, and E. Housos, "Solving the Examination Timetabling Problem in GPUs," pp. 295–327, 2014.
[29]N. Leite, C. M. Fernandes, F. Melício, and A. C. Rosa, "A cellular memetic algorithm for the examination timetabling problem," *Comput. Oper. Res.*, vol. 94, pp. 118–138, 2018.
[30] Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib and Muayed Sadik, 2018. A Modified Algorithm for Improving Lifetime WSN. Journal of Engineering and Applied Sciences, 13: 9277-9282
[31]H. Asmuni, E. K. Burke, J. M. Garibaldi, and B. Mccollum, "Fuzzy Multiple Heuristic Ordering for Examination Timetabling," pp. 334–353, 2005.
[32]N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Appl. Intell.*, vol. 37, no. 1, pp. 1–11, 2012.
[33] Osamah Ibrahim Khalaf, Bayan Mahdi Sabbar''An overview on wireless sensor networks and finding optimal location of node'',Periodicals of Engineering and Natural Sciences, Vol 7, No 3 (2019)
[34] Ayman Dawood Salman1, Osamah Ibrahim Khalaf and Ghaida Muttashar Abdulsahib, 2019. An adaptive intelligent alarm system for wireless sensor network. Indonesian Journal of Electrical Engineering and Computer Science, Vol. 15, No. 1, July 2019, pp. 142~147
[35]Ogudo, K.A.; Muwawa Jean Nestor, D.; Ibrahim Khalaf, O.; Daei Kasmaei, H. A Device Performance and Data Analytics Concept for Smartphones' IoT Services and Machine-Type Communication in Cellular Networks. *Symmetry* **2019**, *11*, 593.