An Intelligent Technique for Solving Timetable Problem

Abdalla El-Dhshan[1], Hegazy Zaher[2], Naglaa Ragaa[3]

[1]*PhD Candidate in Operations Research, faculty of graduate studies for statistical research, Cairo University, Egypt*

[2]*Professor Doctor in Mathematical statistics, faculty of graduate studies for statistical research, Cairo University, Egypt*

[3]*Associate Professor in Operations Research, faculty of graduate studies for statistical research, Cairo University, Egypt*

Abstract: *Timetabling problem is complex combinatorial resources allocation problems. There are two hard and soft constraints to be satisfied. The timetable is feasible if all hard constraints are satisfied. Besides, satisfying more of the soft constraints produces a high-quality timetable. Crow Search Algorithm (CSA) as an intelligence technique presents for solving timetable problem. CSA like all meta-heuristic optimization techniques is a nature-inspire of intelligent behavior of crows. The proposed CSA tested using the well-known benchmark of hard timetabling datasets (hdtt). Taguchi's method used to tune the best parameter combinations for the factors and levels. The tuned parameters of CSA are applied on datasets in separate experiment. The results show that the proposed CSA is superior to generate solutions in reasonable CPU time when compared with other literature techniques.*

Keywords: **Timetable Problem; Metaheuristic Algorithms; Crow Search Algorithm.**

1. INTRODUCTION

Scheduling Timetable is a crucial process for education, manufacturing, transportation, project management, and other various aspects of daily life. Timetable classifies as one of NP-hard problems. This paper predicated on solving the educational timetabling problem as a kind of resource allocation problems. Plentiful of traditional methods presented for building a timetable to prevent waste of time and effort spent on preparing the timetable. The constraints of the timetable problem can be classified into two types. Hard constraints, which never violated, and soft constraints, which should be satisfied but might violate. Generally, Resource allocation of timetable is a process of assigning a set of activities (regular meeting between students and teachers for a specific subject) into a limited number of resources (timeslots of working days and rooms) [1].The objective of timetabling problem is to maximize the quality of a feasible solution. In this work, CSA has presented as a new intelligent method inspired by crows behavior of saving excess food in hiding places and retrieve it when the food is needed. It belongs to metaheuristic population-based approaches, as well as it explores the search space to find a solution with high-quality[2]. The proposed CSA investigated to solve the resource allocation of timetable problem. Hdtt (hard timetable) benchmark is used to examine the efficiency of the proposed CSA comparing with other literature techniques.

This paper consists of seven sections. Section 1 demonstrates a general overview of the research. In section 2, literature review is described. In Section 3, the resource allocation of timetable problem definition is introduced. In Section 4, the crow search algorithm is proposed to solve resource allocation of timetable problem. In Section 5, the proposed CSA for recourses allocation of timetable is described. Section 6 analyzes, testing, and the experimental results of the proposed algorithm. Finally, Section7 includes the conclusion.

2. Literature Review

The optimization process is the way of finding the best solution of a set of feasible solutions with consideration of the problem constraints. Optimization problems have two types of variables discrete or continuous. Discrete optimization problems are named also combinatorial optimization problems [3]. Resource allocation of timetable problem is a combinatorial optimization problem, Artificial intelligence approaches and other optimization techniques are introduced for solving this problem. Simulated annealing [4], tabu search [5],and great deluge [6] techniques of single-based metaheuristic algorithms are introduced for solving this problem type as well as, Genetic algorithms [7]; [8], and artificial bee colony [9] are proposed as population based metaheuristic algorithms. Not only metaheuristic techniques used to solve the problem but also heuristic methods [10] and Integer programming [11] as a traditional approach are used. Although, there are many methods and approaches are adapted to deal with complex optimization problems, no one could efficiently solve all the classes of optimization problems [12].

Recently, one of the newest metaheuristic optimizers, the crow search algorithm (CSA) has been introduced by [2]. The CSA emulates the behavior of crows as intelligent birds. He demonstrated that the CSA has better and competitive results, which were discussed the effectiveness of the technique that can be implemented easily. Furthermore, in another research, [13] used CSA for optimization of the PV/wind/tidal/battery system. In [14] Abdelaziz and Fathy modified the CSA for the optimal design of the radial distribution network by optimal selection of the suitable conductor to minimize the capital and energy loss costs. Liu el al. in [15] proposed an evaluation model of groundwater quality by improved Extreme Learning Machine (ELM) to resolve fuzziness of the water quality evaluation and incompatibility of water parameters. Crow Search Algorithm (CSA) was used to optimize the input weights and thresholds of hidden-layer neurons of the ELM. Mohamadi and Abdi in [16] modified the CSA to solve the non-convex economic load dispatch problem. They proposed an elitist approach for selecting the target crow. Also, they adjusted the flight length of crows based on crows' distance. Rizk-Allah el al. in [17] presented the chaotic crow search algorithm (CCSA) for solving fractional optimization problems (FOPs). They tried to refine the global convergence speed and enhance the exploration/exploitation tendencies. The proposed CCSA has integrated the chaos theory (CT) into the CSA. Javidi el al. in [18] presented an enhanced crow search algorithm for optimum design of the structure. However, there has been less previous evidence for employing the CSA for combinatorial optimization problems, and needs more deep investigations.

3. Resources Allocation of Timetable Problem Definition

Resources allocation has identical entities that are collected and interacted together to achieve a certain goal. Lazarev and Nekrasov in [19] listed the common entities of the resource scheduling problem.

- Activities: Let $A = \{a_1 . a_2 \dots , a_N\}$ is the set of given activities (events, jobs, tasks, etc.) that need a set of resources to process or execute.
- Resources: the physical entities that are used in the scheduling process (e.g. machines, rooms, workers, raw material, etc.). Each resource can process only one task, and the set of all resources designated as R. In case there are M types of resources the set R includes M subsets of typified resources:

$$R = \bigcup_{m=1}^{M} R_m \qquad (1)$$

The timetable is known as a multi-dimensional assignment problem, which combining the shard objects (teacher. class. subject. room. day. and period). The teacher is assigned to teach a subject for a class in a room at a specific time of the day. This assignment has to consider a set of hard and soft constraints.

Let T is a set of teachers.
C is a set of classes.
S is a set of subjects.
R is a set of rooms.
P is a set of periods.

D is a set of days.
$At.s$ is a set of periods when teacher t of subject s is available, and the decision variable is:

$$x_{tcsrpd} = \begin{cases} 1 & if\ teacher\ t\ teaches\ subject\ s\ for\ class\ c\ \ in\ room\ r\ at\ \ period\ p \\ & \qquad\qquad in\ day\ d \\ 0 & otherwise \end{cases} \quad (2)$$

3.1 Hard constraints

The set of hard constraints are the main element of solving the timetable problem. The solution must satisfy all the hard constraints to be a feasible solution. The set of main timetable hard constraints as follows [4].

HC1: A teacher cannot be assigned to more than one class during any time slot.

$$\sum_{c \in C} x_{tcsrpd} \leq 1 \quad \forall\ t \in T_s\ .s \in S.r \in R.p \in P.d \in D. \quad (3)$$

HC2: A class cannot be assigned more than one teacher for any time slot.

$$\sum_{s \in S} \sum_{t \in T_s} x_{tcsrpd} \leq 1 \quad \forall\ c \in C.r \in R.p \in P.d \in D. \quad (4)$$

HC3: A room cannot be allocated more than once for any time slot.

$$\sum_{t \in T_s} \sum_{c \in C} \sum_{s \in S} x_{tcsrpd} \leq 1 \quad \forall\ r \in R.p \in P. \in D. \quad . \quad (5)$$

HC4: A teacher t. teaching subject s. is to be assigned $lt.s$ lectures (periods or time slots) per week.

$$\sum_{c \in C} \sum_{r \in R} \sum_{d \in D} \sum_{p \in P} x_{tcsrpd} = l_{t.s} \quad \forall\ s \in S.t \in T_s\ . \quad (6)$$

HC5: A class c must attend lc lectures per week.

$$\sum_{s \in S} \sum_{t \in T_s} \sum_{r \in R} \sum_{d \in D} \sum_{p \in P} x_{tcsrpd} = l_c \quad \forall\ c \in C. \quad (7)$$

3.2 Soft Constraints

The set of soft constraints used to measure the solution quality. Every problem has a set of soft constraints upon the enterprise rolls such as teacher preference time, Consecutive or isolated lectures, and room utilization.

3.3 Fitness Function

The fitness function is used to evaluate the solution fitness. That is differentiating between the solutions [20].

$$eval(f) = \frac{1}{1 + \text{cost(f)}} \quad (8)$$

That, cost(f) represents the number of violating constraints and it can be calculated as:

$$cost(f) = \sum_{i=1}^{ct} n_i(f) \times w_i \quad (9)$$

Where, ct is the count of constraints. $n_i(f)$is the penalty of violating constraint i and w_i is the weight of constraint i.
The objective is to minimize the count of violated soft constraints for the feasible solution that is already generated by obtaining the hard constraints.

4. Crow Search Algorithm

Crows are extremely intelligent birds. They have a large brain relative to their body size and very close to the human brain based on the brain-to-body ratio. They have many incredible skills such as problem solving and communication. Self-awareness is one of the amazing skills which have the tool-making ability. The researches show that the crow can remember faces and never forget. Furthermore, it has the ability of us helping tools for food searching journey. Besides that, they can remember their hiding places of food for a long time [2].
Crows follow other birds to see where their food will be stored, that to steal the food once it goes. Just it had the food, the crow makes a set of prevarications to deceive other watching birds, they are using their own experience to protect themselves to be a victim later, and to determine the safest actions to avoid their caches from being pilfered.
Based on the above-mentioned, [2] listed the principles of CSA:

- Crows live in the form of a flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.
- Crows protect their caches from being pilfered by a probability.

Let(d), the number of environment dimensions,
N: The number of crows (flock size or population size).
g_{max} : The maximum number of generations.
$xi.g$: The position of crow i at generation g (iteration),
Where, $i = 1.2. \ldots . N;\ g = 1.2. \ldots . g_{max}$
$x^{i.g} = \left[x_1^{i.g}. x_2^{i.g}. \ldots . x_d^{i.g}\right]$ is the vector of search space.
p^{ig}: Represents the best hiding food place obtained by crow i at generation g.

Actually, the crow keeps its best position experience and moves in the environment to find out better food sources (hiding places).
Assume that at generation g, crow j wants to visit its hiding place, p^{ig}. At this generation, crow i decided to follow crow j to discover the hiding place of crow j. in this situation, two cases may be happening:
Case I: Crow j does not know that crow i is following it. Then, crow i will steal the hiding place of crow j. In this case, for the next generation the position of crow i is obtained as follows:

$$x^{i.g+1} = x^{i.g} + r_i \times fl^{i.g} \times \left(p^{ig} - x^{j.g}\right) \quad (\mathbf{10})$$

Where r_i is a random number with a uniform distribution between 0 and 1 and $fl^{i.g}$ is the flight length of crow i at generation g.

Case II: Crow j knows that crow i is following it. Then, to protect its cache from pilfered, crow j will fool crow i by going to another position of the search space.
Generally, the two cases can be summaries as follows:

$$x^{i.g+1} = \begin{cases} x^{i.g} + r_i \times fl^{i.g} \times \left(p^{ig} - x^{j.g}\right) & r_j \geq AP^{j.g} \\ a\ random\ possion & otherwise \end{cases} \quad (\mathbf{11})$$

Where r_j is a random number with a uniform distribution between 0 and 1 and $fl^{i.g}$ is the flight length of crow i at generation g, while $AP^{j.g}$ is the awareness probability of crow j at generation g.

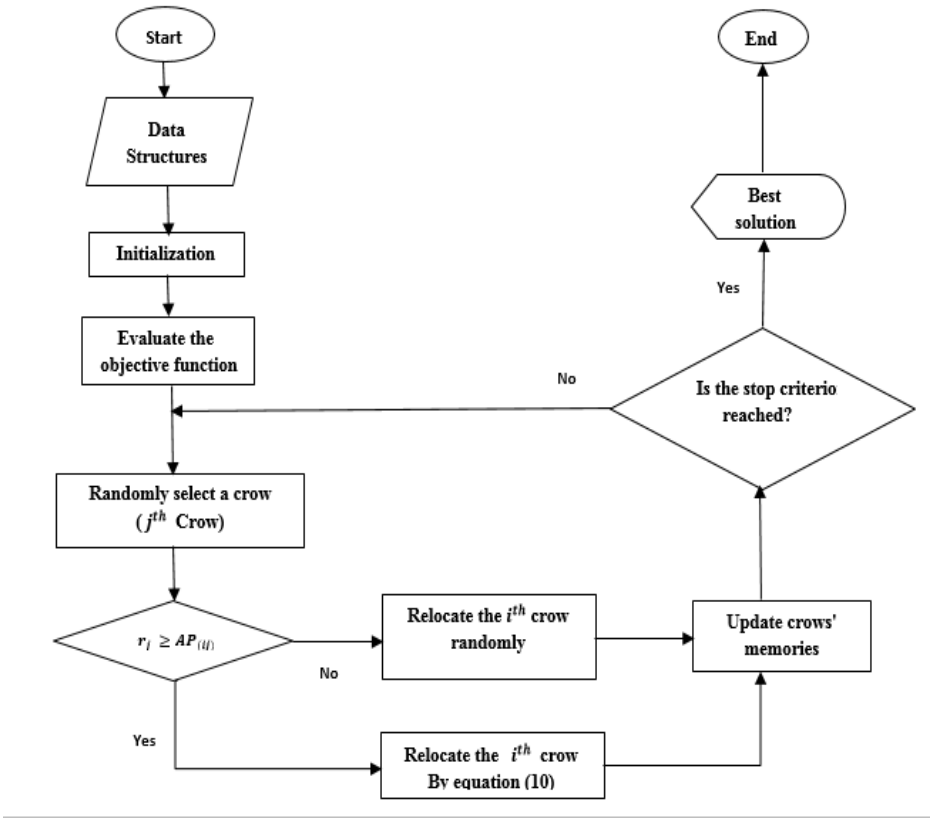

Figure.1 Flowchart of Crow Search Algorithm [21]

5. CSA for Recourses Allocation of Timetable

5.1. Solution Structure

Constructing a reasonable solution structure notably is absent to create a direct relationship between the problem entities (dimensions). Hence, Working with multidimensional problem is very complicated to solve. Therefore, it is useful to associate the related entities into two-dimensional.

Timetable as a resources allocation problem can be abstracted into two main entities.The first main entity is the available resources $S = \{s_1. s_2. \ldots . s_j\}$, which contains the combination of period's $P = \{1.2. \ldots . p\}$. rooms $R = \{1.2. \ldots . r\}$. and daysD $= \{1.2. \ldots . \mathrm{d}\}$. The combination can represent as a vector with length (p × r × d). The second dimension is activities, that can be represented as $A = \{a_1. a_2. \ldots . a_i\}$. this dimension concatenates subjects $\mathrm{S} = \{1.2. \ldots . \mathrm{s}\}$. teachers $\mathrm{T} = \{1.2. \ldots . \mathrm{t}\}$. and classes (group of students) $\mathrm{C} = \{1.2. \ldots . \mathrm{c}\}$.

Figure 2 gives an illustration of solution structure, the activities are ordered as the dimension of the allocation array, and the resource integer encoding sj will assign to the activity ai. The proposed solution structure is not only used for representing the problem dimensions, but it avoids violating the first three hard constrains of the problem which described in equations (3, 4, and 5) [8].

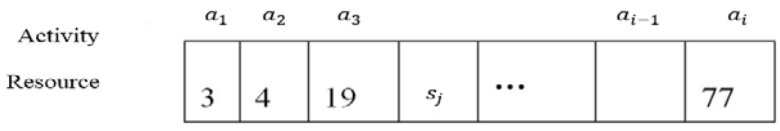

Figure .2 Solution Structure

Example:

Assume that there are 2 rooms, 3 days, and 2 periods per day, And there are 10 activities need available resource to allocate. Then, the number of the available resources will be:

$$= 2\ (\text{rooms}) \times 3(\text{days}) \times 2(\text{periods})$$
$$= 12\ \ (\text{avilible resources})$$

- The activity list:

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A9	A 10

- The available resources list:

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12

The resource will select randomly from list of the available resources to allocate. Starting from the first activity A1, select random resources let it S4 and remove S4 from available resource. Then, S4 allocate to activity A1. The second activity, select random resource let it S7 and remove S7 from available resource, then Resource S7 allocate to activity A2 and the same procedure applies to all activity to get the final solution structure.

- Solution Structure

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A9	A 10
S4	S7	S9	S12	S3	S6	S1	S10	S8	S5

- The reminder available resources list:

~~S1~~	S2	~~S3~~	~~S4~~	~~S5~~	~~S6~~	~~S7~~	~~S8~~	~~S9~~	~~S10~~	S11	~~S12~~

5.2. The Initial Population

As mentioned before in problem definition, there are group of activities (Teacher, Class and Subject) required a suitable resource (Room, Day and Period) to create the crows (solutions). For each new crow, the proposed algorithm is ordered as list of activities then, it is started with the first event selecting randomly resource from the available resources, and then, the selected resource allocates to the current activity. Figure 1 illustrates the basic framework of CSA, and the steps of the CSA are stated as follows [18]:

Crow Search Algorithm

Step 1: Initialize algorithm parameters: Assign number of population(n), maximum number of generation g_{max}, flight length (fl) and awareness probability (AP).

Step 2: Generate initial solutions randomly, check feasibility and fill memory

$$Population = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix}$$

$$Memory = \begin{bmatrix} m_1^1 & m_2^1 & \dots & m_d^1 \\ m_1^2 & m_2^2 & \dots & m_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_1^n & m_2^n & \dots & m_d^n \end{bmatrix}$$

Where d is number of decision variable (problem dimension).

Step 3: Calculate objective function (fitness function) by equation (9).

Step 4: Generate new solution by equation (11).

Step 6: Calculate objective function of new solutions like in step3.

Step 7: Update the memory: If the new solution is better than its best, the memory will be updated. Otherwise, the memory should not be updated:

Step 8: Repeat steps 4 to 7 until the termination criterion is satisfied and retune the best global solution for all population.

5.3. Reproduction

The proposed approach depends only on the mutation operator for generating the new population (crows) from its main crow source. The previous work [8] has experimented 16 type of mutation on the hdtt instances. It used the Relative Percentage Deviation (RPD) to select the fittest mutation type. This mutation is based on generating a random number for $\mathrm{NS} \in [1.\mathrm{a}]$. where a is the length of the available events. Then, the basic two point swap is performed NS times. The NS is calculated as following:

$$NS = Int\big(Rand(1.a)\big) \quad (\mathbf{12})$$

6. Experimental Results

The datasets bank of OR Library provides timetable datasets called HDTT. All Instances and its description are available and share the official URL website1. HDTT denotes to the instances are hard timetabling problems. All periods must be used with very little or no options for each allocation. Each week has a five working days and each day has a six periods with 30 timeslots are available per room.

CSA implemented using Visual Studio 2010 in C# language. Output results of all runtimes have measured using the Intel Core 7 and 8GB RAM

The proposed CSA approach works through factors, which have to be tuned to produce better solutions in reasonable. These factors are the population size (P_{Size}), the number of generations (G_{max}), and the awareness probability value (AP). The factors and their levels are summarized in the following table:

Table1. Factors and levels of the CSA

Factor	Levels
P_{Size}	5, 10, and 15
G_{max}	10000, 20000, and 30000
AP	0.05, 0.10, and 0.15

In Table1, the values of the factors are determined empirically to fit the running time consumption. The experiments number that should be executed to solve each problem according to the aforementioned factors and levels are equal to $3^3 = 27$ experiments. Due to the cost and the time of experiments result, Taguchi's method will be used to determine the best combinations of the factors and levels.

Taguchi's method begins by selecting the predesigned orthogonal array by calculating the total degree of freedom. In this experimentation, one degree of freedom for the total mean, and two degrees of freedom for the each factor. Therefore, the total degree of freedom is equal to $1 + (3 \times 2) = 7$. Thus, the appropriate predesigned orthogonal array at least has 7 rows.

Table 2. Orthogonal Array

No.	P_{Size}	G_{max}	CR	T
1	5	10000	0.0003	10000
2	5	20000	0.0005	15000
3	5	30000	0.0007	20000
4	10	10000	0.0003	10000
5	10	20000	0.0005	15000
6	10	30000	0.0007	20000
7	15	10000	0.0003	10000
8	15	20000	0.0005	15000
9	15	30000	0.0007	20000

Table 2 shows the available predesigned orthogonal array $L_9(3^3)$, where the factors are assigned to columns and levels are assigned to rows.
In order to determine the relative importance of each factor, Analysis of Variance for Means are used. According to the orthogonal array, 9 trails have been executed on instance hdtt8 which is the most difficult instance in the dataset. Every trail has run 5 times and recorded the cost and CPU time as responses. After applying the experiment, hence, the cost responses for all trails yield zero cost, so, CPU time used as unique response. In this experiment, Minitab program version 17.1.0 used to design the Taguchi experiment. Since, it is a good tool using to design like this experiment. In addition to, there are a variety number of options that are supported to analysis the data.

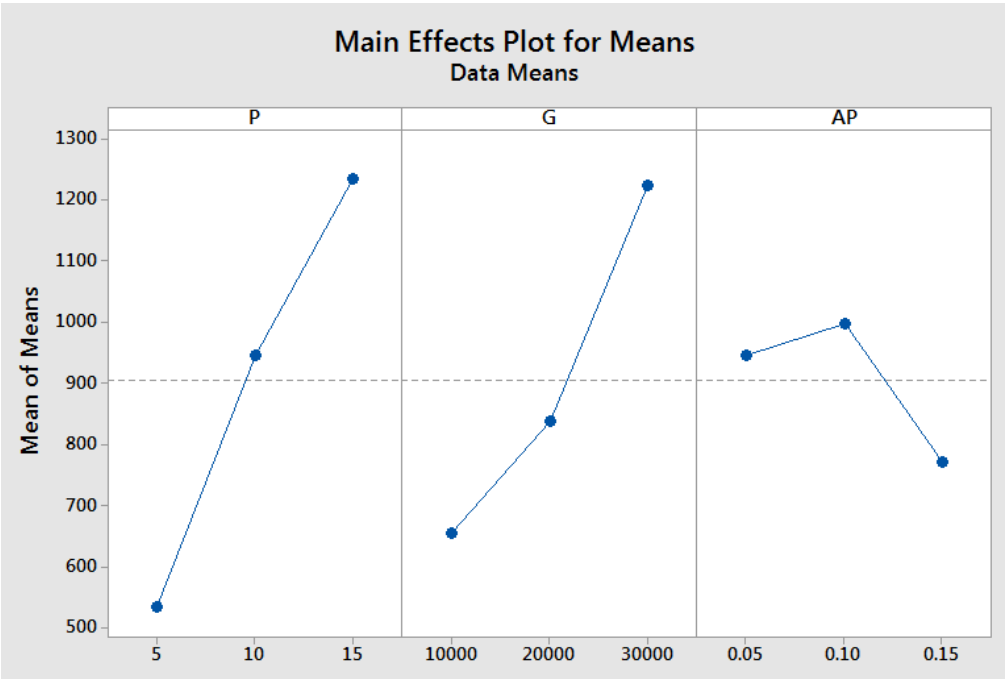

Figure 3. Main effects plot for means

Table 3. Rank of the factors for Means responses

Level	P_Size	G_Max	AR
1	535.3	655.1	946.5
2	946.4	838.6	998.3
3	1235.9	1223.9	772.8
Delta	**700.6**	**568.8**	**225.5**
Rank	**1**	**2**	**3**

Table 3 illustrates the rank of the factors that are used in the experiment, which indicates the number of population size is the important factor of the other factors. The generation number factor follows the generation number then the awareness probability.
The best values of the levels recommended to be as follows:

Table 4. The best level for each factor

Factor	Level
P_{Size}	5
G_{max}	10000
AR	0.15

The tuning parameter values are used for solving Course Timetable Problem CTTP are (5) population size. (10000) generation number. (0.15) awareness probability (AP). The proposed CSA are compared with other literature methods used to solve the timetable problems for the same benchmark instances, which described as follows:

- **SA1** Simulated Annealing Algorithm [22].
- **SA2** Simulated Annealing Heuristic [23].
- **TS** stand to Tabu Search. GS a Greedy Search. NN-T2 and NN-T2 are Hopfield Neural Networks [24].
- **DWTAN** and CPMF are neural network approaches [25].
- **SA3** is a Simulated Annealing variant [26].
- **TFH** Timeslot-filling heuristic and EAH event-assignment heuristic. The both of them are a heuristic methods described in [10].
- **GAHCO** denotes to Genetic Algorithm with Hill Clamping Optimization [8].
- **CSA** denotes to the proposed crow search algorithm.

Table 5 compares the results of CSA with other methods. The first column indicates the count of violated constraints for the best result achieved, and the second column for the average of violated constraints of 20 runs. The results show that the CSA is the best of all methods, which yields zero cost as the best cost and the average cost values for all instances. In addition, no cost all datasets. The performance of the proposed algorithm is competitive when compared to the other methods, and performs the best finding timetables for all 20 runs conducted for each data set.

7. Conclusion

This study proposes the crow search algorithm CSA for solving the resources allocation of timetable problem. The results indicate that the crow search algorithm provide a good potential mechanism compared with other mechanisms. CSA gives optimal solutions with one hundred percentages of fitness values within a few seconds for all datasets of the benchmark. So, the proposed approach can add as a new method for solving resources allocation problems. The

future work can investigate used the proposed CSA for further applies to similar combinatorial optimization problems could prove quite beneficial.

Table 5. Comparative result of the CSA with other methods

Method	HDTT4		HDTT5		HDTT6		HDTT7		HDTT8	
	Best Cost	Avg. Cost	Best Cost	Avg. Cost	Best Cost	Avg. Cost	Best Cost	Avg. Cost	Best Cost	Avg. Cost
GS	5	8.5	11	16.2	19	22.2	26	30.9	29	35.4
CPMF	5	10.7	8	13.2	11	18.7	18	25.6	15	28.6
TS	**0**	0.2	**0**	2.2	3	5.6	4	10.9	13	17.2
EAH	**0**	**0**	2	5.4	6	7.9	9	12	13	15
DWTAN	**0**	**0**	**0**	0.4	**0**	1.65	**0**	2.1	**0**	3.25
TFH	**0**	**0**	**0**	0.6	**0**	2.1	**0**	2.5	**0**	3.1
SA1	-	-	**0**	0.7	**0**	2.5	2	2.5	2	2.5
SA2	**0**	**0**	**0**	0.3	**0**	0.8	**0**	1.2	**0**	1.9
NN-TT2	**0**	0.1	**0**	0.5	**0**	0.8	**0**	1.1	**0**	1.4
NN-TT3	**0**	0.5	**0**	0.5	**0**	0.7	**0**	1	**0**	1.2
GAHCO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0.6
SA3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0.4
CSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

REFERENCES

[1] A. Eludire and C. Akanbi, “A Conceptual Approach to Resources Allocation Scheduling,” *J. Adv. Math. Comput. Sci.*, vol. 25, no. 6, 2017, doi: 10.9734/jamcs/2017/32569.

[2] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm,” *Comput. Struct.*, vol. 169, 2016, doi: 10.1016/j.compstruc.2016.03.001.

[3] J. S. Arora, M. W. Huang, and C. C. Hsieh, “Methods for optimization of nonlinear problems with discrete variables: A review,” *Structural Optimization*, vol. 8, no. 2–3. 1994, doi: 10.1007/BF01743302.

[4] D. Zhang, Y. Liu, R. M’Hallah, and S. C. H. Leung, “A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems,” *Eur. J. Oper. Res.*, vol. 203, no. 3, 2010, doi: 10.1016/j.ejor.2009.09.014.

[5] Z. Lü and J. K. Hao, “Adaptive Tabu Search for course timetabling,” *Eur. J. Oper. Res.*, vol. 200, no. 1, 2010, doi: 10.1016/j.ejor.2008.12.007.

[6] K. Shaker and S. Abdullah, “Incorporating great deluge approach with kempe chain neighbourhood structure for curriculum-based course timetabling problems,” 2009, doi: 10.1109/DMO.2009.5341894.

[7] R. Raghavjee and N. Pillay, “A comparison of genetic algorithms and genetic programming in solving the school timetabling problem,” 2012, doi: 10.1109/NaBIC.2012.6402246.

[8] M. M., R. A., and A. M., “Genetic Algorithm for Solving Course Timetable Problems,” *Int. J. Comput. Appl.*, vol. 124, no. 10, 2015, doi: 10.5120/ijca2015905408.

[9] A. L. aro Bolaji, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, “University course timetabling using hybridized artificial bee colony with hill climbing optimizer,” *J. Comput. Sci.*, vol. 5, no. 5, 2014, doi: 10.1016/j.jocs.2014.04.002.

[10] M. Pimmer and G. R. Raidl, “A Timeslot-Filling Heuristic Approach to Construct High-School Timetables,” 2013.

[11] A. E. Phillips, H. Waterer, M. Ehrgott, and D. M. Ryan, “Integer programming methods for large-scale practical classroom assignment problems,” *Comput. Oper. Res.*, vol. 53, 2015, doi: 10.1016/j.cor.2014.07.012.

[12] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, 1997, doi: 10.1109/4235.585893.

[13] A. Askarzadeh, “Electrical power generation by an optimised autonomous PV/wind/tidal/battery system,” *IET Renew. Power Gener.*, vol. 11, no. 1, 2017, doi: 10.1049/iet-rpg.2016.0194.

[14] A. Y. Abdelaziz and A. Fathy, “A novel approach based on crow search algorithm for optimal selection of conductor size in radial distribution networks,” *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 2, 2017, doi: 10.1016/j.jestch.2017.02.004.

[15] D. Liu *et al.*, “ELM evaluation model of regional groundwater quality based on the crow search algorithm,” *Ecol. Indic.*, vol. 81, 2017, doi: 10.1016/j.ecolind.2017.06.009.

[16] F. Mohammadi and H. Abdi, “A modified crow search algorithm (MCSA) for solving economic load dispatch problem,” *Appl. Soft Comput. J.*, vol. 71, 2018, doi: 10.1016/j.asoc.2018.06.040.

[17] R. M. Rizk-Allah, A. E. Hassanien, and S. Bhattacharyya, “Chaotic crow search algorithm for fractional optimization problems,” *Appl. Soft Comput. J.*, vol. 71, 2018, doi: 10.1016/j.asoc.2018.03.019.

[18] A. Javidi, E. Salajegheh, and J. Salajegheh, “Enhanced crow search algorithm for optimum design of structures,” *Appl. Soft Comput. J.*, vol. 77, 2019, doi: 10.1016/j.asoc.2019.01.026.

[19] A. Lazarev and I. Nekrasov, “Mathematical models for enterprise resource scheduling: Complexity of key approaches to problem formulation,” 2017, doi: 10.1109/MLSD.2017.8109650.
[20] M. Doulaty, M. R. F. Derakhshi, and M. Abdi, “Timetabling: A State-of-the-Art Evolutionary Approach,” *Int. J. Mach. Learn. Comput.*, 2013, doi: 10.7763/ijmlc.2013.v3.314.
[21] B. Zolghadr-Asli, O. Bozorg-Haddad, and X. Chu, *Crow Search Algorithm (CSA). In Advanced optimization by nature-inspired algorithms.* Singapore: Springer, 2018.
[22] D. Abramson and H. Dang, “School Timetables: A Case Study in Simulated Annealing,” in *Lecture Notes in Economics and Mathematical Systems*, Berlin, Heidelberg: Springer, 1993, pp. 103–124.
[23] M. Randall and D. Abramson, “A general meta-heuristic based solver for combinatorial optimisation problems,” *Comput. Optim. Appl.*, vol. 20, no. 2, 2001, doi: 10.1023/A:1011211220465.
[24] K. A. Smith, D. Abramson, and D. Duke, “Hopfield neural networks for timetabling: Formulations, methods, and comparative results,” *Comput. Ind. Eng.*, vol. 44, no. 2, 2003, doi: 10.1016/S0360-8352(02)00180-8.
[25] M. P. Carrasco and M. V. Pato, “A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem,” in *European Journal of Operational Research*, 2004, vol. 153, no. 1, doi: 10.1016/S0377-2217(03)00099-7.
[26] M. Gendreau and J. Y. Potvin, “Metaheuristics in combinatorial optimization,” *Ann. Oper. Res.*, vol. 140, no. 1, 2005, doi: 10.1007/s10479-005-3971-7.