2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)

Integrating ABC with genetic grouping for university course timetabling problem

Elham Ghasemi,
Department of computer engineering,
University of Kurdistan,
Sanandaj, Iran,
ghaseme.elham@gmail.com

Parham Moradi,
Department of computer engineering,
University of Kurdistan,
Sanandaj, Iran,
p.moradi@uok.ac.ir

Mohammad Fathi
Department of computer engineering,
university of Kurdistan,
Sanandaj, Iran,
mfathi@uok.ac.ir

***Abstract*— scheduling courses in university is an important matter in all academic institutes across the world. Scheduling courses, students, and class rooms without any crash is the main aim of university course time tabling problem. This problem is categorized as a NP-hard problem. The proposed algorithm is firstly based on a genetic grouping approach to generate feasible solutions. In the second step, an effective neighborhood structure which is embedded in an artificial bee colony is used to overcome the problem's conflicts. Experimental results showed that proposed algorithm can obtain comparative results with the best known results of previous articles. The proposed algorithm has been performed on a standard and well known dataset named Socha. The results revealed the efficiency of proposed method. The suggested approach could find the best results on large scale instances of Socha dataset. Results on medium size of the dataset has been improved approaches in four cases out of five instances of dataset.**

Keywords—Neighborhood structure, soft constraint, university course timetabling, Socha dataset

I. Introduction

The first research related to university course timetabling problem has been done in six decades ago by Gotlieb [1]. Managing courses, students and class rooms with no conflict is the main aim of university course timetabling problem [2]. Scheduling courses is a hard and time consuming activity for academic institutes, hence they have been tried to find automatic and optimized approach to solve this problem. Graph coloring and constraint based methods [3] are one of the oldest approach to solve this problem [4]. One of the most well-known methods to overcome the problem is genetic algorithm [5-9]. New researches which used genetic algorithm are combining this algorithm with some optimizer methods like hill climbing and local search [6]. Genetic grouping is also an effective way to find a feasible solution for the problem [10]. Mathematics methods like logic programing is another manner for solving this problem [11, 12]. Hybrid harmony search [13] and artificial bee colony [14, 15] are two new methods which obtained prominent results. Simulated annealing [16-19] and Tabu search [20, 21] are metaheuristic approach which has been used to tackle the problem with fairly results. Artificial bee colony is another effective algorithm to overcome the problem. ABC which was introduced by Karaboga [22] widely used to solve complicated problem in artificial intelligence [15] like nurse rostering problem [23] and multi robots stick carrying [24].

In this paper a novel genetic grouping approach is used to find the feasible solutions. In the second phase, a proposed artificial bee colony is applied to enhance the quality of feasible solutions. A new neighborhood structure is embedded in artificial bee colony to make it more effective. The next parts of the paper are categorized as follows: session II is a description of problem, session III illustrated the proposed genetic group algorithm and session IV are provided bee colony algorithm and final part represents experimental results.

II. Problem Description

A. University course timetabling problem

University course timetabling problem is a NP-hard problem [6, 10, 25]. In this problem the major issue is scheduling some courses into some period of times and class rooms [26]. Each class room has a specific capacity so that the number of enrolled students should be equal or lesser than class room capacity. Otherwise, a hard conflict would be occurred [27]. A solution without any hard conflict is considered as a feasible solution. Equation (1) shows a feasible solution which S is showing a candidate solution. Yet there is a fitness function description in equation (2). The aim is finding a solution which is free from hard violation and lowest number of soft constraints. Hard and soft constraints is described in the next part.

$$fesibility(p) = \sum_{p=1}^{n} \text{hard constraint (Sp)} = 0 \qquad (1)$$

$$fitness(p) = \sum_{p=1}^{n} \text{SoftConstraint (Sp)} \qquad (2)$$

None of proposed methods up to now, could find a solution on medium and large scale instances of Socha dataset which is free from soft violation. Hence, all proposed methods in this domain have been tried to decrease soft constraints [13, 18]. Most of standard papers related to university course timetabling problem have been used Socha dataset. This dataset consists of 5 small size, 5 medium size, and 1 large size instances [28]. In this paper, medium and large instances of Socha dataset are considered in experimental results. Table I shows the dataset parameters. The number of courses, classrooms, features and students in three different sizes is shown in the table.

78-1-4673-9280-8/15/$31.00 ©2015 IEEE

Table I. Socha dataset instances

	Courses	Rooms	Features	Students
Easy	100	5	5	80
Medium	400	10	5	200
Hard	400	10	10	400

This paper applied a two steps strategy to conquer the problem. First step is used a proposed genetic grouping algorithm for removing the violation of hard constraints. Second step have used artificial bee colony (ABC) to decrease violating of soft constraints. The proposed ABC has been applied a neighborhood to improve the results.

B. Constraints

There are six constraints in university course timetabling problem [29]. The first and most important constraints which is covering three out of six constraints are known as hard constraints. In order to achieve a possible solution, all of hard constraints must be omitted. For instance, a student cannot participate in two class in the same time. Removing soft constraints are not compulsory; although, eliminating them as much as possible is desirable and cause to enhance the quality of solution. For example having more than two courses in one specific day, is a kind of soft constraints.

Table II. Constraints of problem

Hard Constraints	i.	Inequality between class room's capacity and students number
	ii.	Conflict between two courses in a same class room
	iii.	Attending a student simultaneously in two courses
Soft Constraints	i.	More than two courses in one day for each student
	ii.	Sole course in a day for each student
	iii.	A courses in last timeslot of a day for each student

III. PROPOSED GENETIC GROUPING

A. Chromosome structure based on genetic grouping

University course timetabling problem (UCTP) has a vast search space; hence a method to categorize similar items in order to minimize the search space is crucial. Genetic grouping is an appropriate way to obtain the mentioned aim. Generally, grouping problem is allocating a number of items into a collection, in a manner that no common member exists in two separate groups.

Chromosome structure is demonstrated in figure 1. To overcome the hard constraints which are categorized in table II, a structure based on genetic grouping is used. Each chromosome which have been considered as a solution, is contains 45 genes. In order to obtain a feasible solution for this problem, Courses should be placed appropriately in the genes without any conflict. Each time slot is a time group. There are 45 time groups in the chromosome.

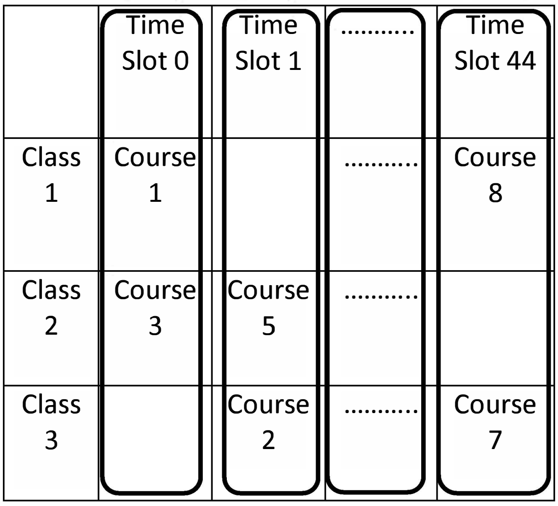

Figure 1. Chromosome Representation

There are 5 days on an academic week and 9 time slots per each day. The rows are class rooms and columns represent the time slots. All the rooms have specific capacity.

Courses in each gene must have smaller or equal capacity of rooms' capacity. The capacity of course is defined by the number of students who enrolled in that course. Subsequently, the first hard constraint that is defines in table II is removed. Algorithm 1 shows the pseudo code of genetic grouping. All of genes in the chromosome are permitted to have a unique course; so, conflict between two courses in a same room which is the second hard constraint in table II won't be met. The most crucial hard constraints is attending simultaneously in two courses by one specific student.

B. crossover in genetic grouping

In the crossover, four random time groups should be chosen in two stochastic chromosomes randomly. Points a, and b from the first chromosome should be replace by points c and d in second chromosome. After that a checking must be done to avoid of bringing repetitive courses in a chromosome. Otherwise, duplicate courses must be omitted and replaced by a non-repetitive course.

C. Mutation Operater

In mutation operator, a solution which has the best fitness value from initial solutions is chosen; after that the solution is checked to obtain number of conflict between courses. A replacement should be done for the course which has the most conflict in a solution. A time group with lowest number of courses is chosen for replacing the mentioned course. After mutation as shown in algorithm 1, a proposed neighborhood structure that is described in the next sections is applied in genetic grouping. The aim of using proposed genetic grouping is finding feasible solutions for the problem. After obtaining feasible solutions, a proposed method based on artificial bee colony is applied to improve the quality of solutions.

IV. PROPOSED METHOD BASED ON ARTIFICIAL BEE COLONY

To defeat the soft constraints, a novel algorithm based on artificial bee colony (ABC) is proposed in this paper [30, 31]. Employee bees, onlooker bees and scout bees are the main parts in ABC algorithm. Our proposed method is concentrated on a novel neighbor structure.

A. Employee Bee

In the first phase of artificial bee colony (ABC) which is known as employee bee, after producing initial population the fitness function according to formulate 2 is calculated for all produced solutions. The most important aim in ABC algorithm is reducing the fitness function which is showed in formulate 2. On the other words, the soft constraints should be decreased as much as possible. It is worth pointing out that, having a sole course in a day for a student is the most controversial one between three soft constraints in table II. A proposed neighborhood structure is applied in employee bee phase in order to decrees the number of soft constraints.

Proposed neighborhood structure

First of all, a list of courses which are sole in a day for a student should be defined. The proposed neighborhood structure is used both in genetic grouping and employee bee phase of ABC algorithm. It is important to say that, the random time slot must be chosen in the same row of mentioned course. Because each row is defining a room with certain capacity, and changing a course into a different row would cause Inequality between class room's capacity and student's number. If the number of fitness function get decrease, then the new change will be accepted.

Table III. Parameter setting

Parameter	Value
Colony size	10
Iteration of genetic grouping	5000
Iteration of ABC	2000
Exhausted limitation	100
Crossover rate	0.4
Mutation rate	0.2
Proposed neighborhood rate	0.7

- Neighborhood 1

Finding the places in timetable which are contain a sole course for a pupil, is the first step in neighborhood 1. After that, the location of the course in mentioned place should be changed randomly.

- Neighborhood 2

In this type of neighborhood, all the rows in a chromosome are divided into two equal parts and the courses of one part would be exchange randomly. Obviously, courses with high degree of violating constraints are in priority. In the chosen part, two different groups would be selected randomly and courses in each group should be exchanged as illustrated in figure 2. Each course can only change place in its row; because of room's capacity issue. Simultaneously checking for avoiding hard constraint existence is crucial.

- Neighborhood 3

Two random selected time groups would exchange their courses in neighborhood 3. To implement this neighborhood, two time groups should be chosen, so that one of them which have the most conflict in the timetable is selected firstly. Second time group should have the lowest number of courses. Algorithm 2 shows the pseudo code of proposed neighborhood.

```
1.  While h(x) > 0
2.  Create a random chromosome (genetic grouping)
3.  Crossover neighborhood
4.  Mutation
5.  Proposed neighborhood
6.      If (h(x))= 0
7.  Save the created chromosome
8.      End If
9.  Else
10. Go to 1
11. Until (iteration= 0)
12.     Else If (h(x)= 0)
13. Go to 6
14.     End If
15. End while
```

Algorithm 1. Initial feasible solution based on genetic grouping

B. Onlooker Bee

Solutions obtained from previous section are sorted according their fitness values; after that, one solution is selected by roulette wheel. Then a neighbor is generated by mentioned structure and solutions that don't enhanced after 10 trials would be exhausted.

C. Scout bee

If the new population produced by proposed neighborhood was better than previous one, then the new one will be accepted. Otherwise solutions which didn't optimized would be remove after 100 trials. To replace the exhausted population with new one, scout bees would performed this duty.

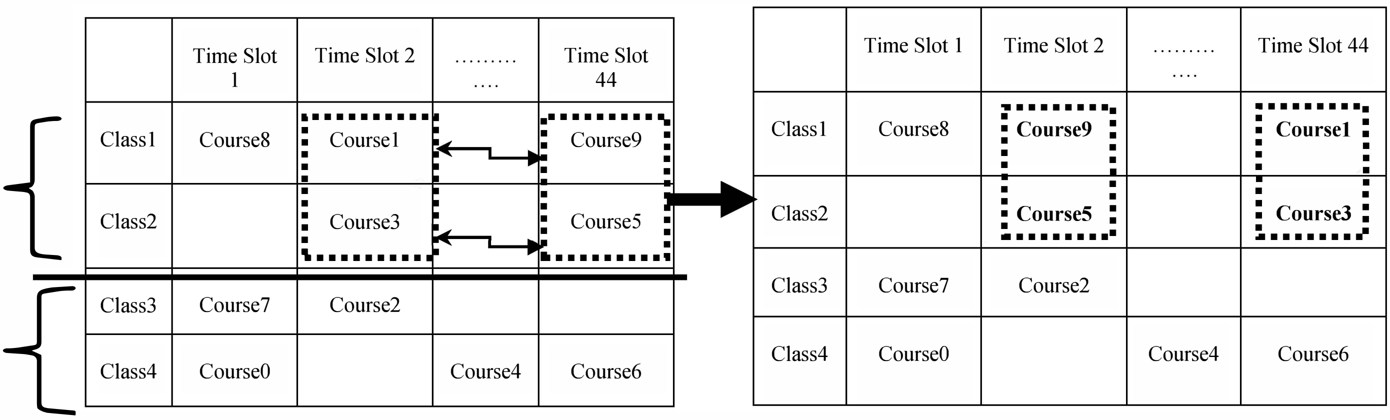

Figure 2. Neighborhood 2

```
Pop= Select a feasible solution
N= Choose a random number (between 1-3)
  If N= 1
       New pop= neighborhood 1
Else
     If N= 2
        New pop = neighborhood 2
  Else
      If N= 3
          New pop = neighborhood 3
      End if
    End if
  End if
  If fitness(New pop) < fitness(Pop)
& feasibility= True then
    Pop= New pop
  End if
End
```

Algorithm 2. Proposed neighborhood

Algorithm 3 shows the complete algorithm which is a combination of genetic grouping and artificial bee colony to overcome the problem.

```
Algorithm 1
Employee bee
Algorithm 2
Onlooker bee
Scout bee
   If h(x) and s(x)= 0
End
   Else
Go to 1
```

Algorithm 3. Proposed Method

V. EXPERIMENTAL RESULT

Table III shows the setting of parameters which are used in the article. These values have been obtained based on several run to find the best value for each parameter. Crossover and mutation rate are used in genetic grouping to find feasible solution. The proposed algorithm have been tested on a PC with Intel 2.2 GHz and 4GB RAM. The best solution for each problem instances highlighted in bold.

A. Parameters setting

Experiments revealed that setting crossover and mutation rate to 0.4 and 0.2 respectively, can obtain best result. A rate is considered for proposed population; this rate would indicated that how many percent of population get affected by neighborhood structure. According to experiments, the best rate for this parameter is 0.7 percent of whole population. Parameter setting is presented in table III.

B. Comparative results

Table IV is demonstrating the comparative results on 10 runs between five different algorithms with our proposed approach on medium and large size instances of Socha dataset. The comparisons are among GSGA [6] which is a Genetic Algorithms with Guided and Local Search Strategies. Hybridized-ABC [15] is an algorithm based on hybridizing artificial bee colony with hill climbing optimizer, HHSA is a combination of harmony search algorithm That has been hybridized by hill climbing, with a particle swarm optimization [29], ACS-TS [32] that is a combination of ant colony algorithm and Tabu search, ENGD [33] which is based on non-linear great deluge, and the proposed algorithm in this article which is based on a proposed genetic grouping and a combination of ABC with a novel neighborhoods structure. Table IV shows the results of our proposed method and five different algorithms. Lower fitness value is a better value for the UCTP. The best results are bold in table IV.

From table V It can be seen that our proposed method can achieve lower average over ten runs in term of best, mean, worst and standard deviation on the 6 instances of Socha dataset. It can be seen that our proposed method can achieve lower average in all cases except M4 and M2 instances of dataset. The best value for most of the instances has been achieved by our method are the best among five other algorithms. Obviously, the reported results from our proposed method are better from other approaches. This results proves that proposed algorithm is more effective; because of using a novel genetic grouping and ABC algorithm. According to reported result, our proposed method can find results on large instance of Socha dataset which are considerably better than previous algorithms. On medium instances of Socha dataset, results are better in 4 cases except one instances in comparison to five other algorithms.

Table IV. Comparison results

	Hybridized-ABC	GSGA	HHSA	ENGD	ACS-TS	**Proposed Algorithm**
Medium 1	73	152	99	126	150	**65**
Medium 2	79	101	73	123	179	**70**
Medium 3	137	121	130	185	183	**90**
Medium 4	69	98	105	116	140	**58**
Medium 5	61	103	**53**	129	152	59
Hard 01	462	645	385	821	750	**299**

VI. CONCLUSION

In this paper an algorithm based on genetic grouping and artificial bee colony has been proposed to overcome university course timetabling problem. Genetic grouping is used to find initial feasible solution.

Artificial bee colony as an effective approach to enhance the quality of feasible solution has been used. A novel neighborhood structure as an effective strategy has been applied to genetic grouping and artificial bee colony. Experiments performed on medium and large size instances of Socha dataset. The results revealed that proposed method can find solutions with considerable enhancement in performance over previous methods. Future work will be applied the proposed method on exam timetabling problem. The performance of proposed method can be enhanced by combining the artificial bee colony with other type of evolutionary algorithms like memetic algorithm.

Table V. Results on the best, worst, average and standard deviation for proposed method and three other algorithms

Dataset instance	Algorithm	Best	Average	Worst	Std
M1	Proposed method	85	**93.3**	114	17.30
	Hybridized-ABC	**73**	111.70	143	19.21
	GSGA	240	247	260	9.02
	HHSA	98	137.8	155	12.57
M2	Proposed method	**70**	108.4	169	10.021
	Hybridized-ABC	73	**102.50**	118	14.98
	GSGA	162	172.4	209	14.49
	HHSA	102	121.3	138	11.3
M3	Proposed method	**102**	**117.75**	123	26.60
	Hybridized-ABC	137	171.10	211	27.29
	GSGA	242	247	290	6.021
	HHSA	178	199.5	216	11.05
M4	Proposed method	**58**	92.7	110	17.08
	Hybridized-ABC	69	**92.50**	107	12
	GSGA	158	162.7	212	17.01
	HHSA	112	131	143	8.68
M5	Proposed method	**59**	**74.5**	109	15.96
	Hybridized-ABC	101	87.00	128	23.81
	GSGA	124	128.5	200	23.67
	HHSA	77	86.2	94	6.53
Hard1	Proposed method	**299**	**351.7**	464	44.18
	Hybridized-ABC	462	509.70	601	38.07
	GSGA	801	858.2	921	40.35
	HHSA	435	464	482	15.75

REFERENCES

[1]. C. C. Gotlieb, T.c.o.c.-t.t., " in Proc. and C.M.P. IFIP Congr., Ed., 1962, pp. 73–77.

[2]. Lewis, R. and J. Thompson, Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. European Journal of Operational Research, 2015. 240(3): p. 637-648.

[3]. Pongcharoen, P., et al., Stochastic Optimisation Timetabling Tool for university course scheduling. International Journal of Production Economics, 2008. 112(2): p. 903-918.

[4]. Burke, E.K., et al., A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research, 2007. 176(1): p. 177-192.

[5]. Mahiba, A.A. and C.A.D. Durai, Genetic Algorithm with Search Bank Strategies for University Course Timetabling Problem. Procedia Engineering, 2012. 38(0): p. 253-263.

[6]. Shengxiang, Y. and S.N. Jat, Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2011. 41(1): p. 93-106.

[7]. Erben, W. and J. Keppler, A genetic algorithm solving a weekly course-timetabling problem, in Practice and Theory of Automated Timetabling, E. Burke and P. Ross, Editors. 1996, Springer Berlin Heidelberg. p. 198-211.

Abdullah, S. and H. Turabieh. Generating university course timetable using genetic algorithms and local search. in Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on. 2008. IEEE.

[8]. Kohshori, M.S., M.S. Abadeh, and H. Sajedi. A fuzzy genetic algorithm with local search for university course timetabling. in Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on. 2011. IEEE.

[9]. Lewis, R. and B. Paechter, Finding feasible timetables using group-based operators. Evolutionary Computation, IEEE Transactions on, 2007. 11(3): p. 397-413.

[10]. Guéret, C., et al., Building University timetables using constraint logic programming, in Practice and Theory of Automated Timetabling, E. Burke and P. Ross, Editors. 1996, Springer Berlin Heidelberg. p. 130-145.

[11]. Lajos, G., Complete University modular timetabling using constraint logic programming, in Practice and Theory of Automated Timetabling, E. Burke and P. Ross, Editors. 1996, Springer Berlin Heidelberg. p. 146-161.

[12]. Fong, C.W., et al., A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems. Information Sciences, 2014. 283: p. 1-21.

[13]. Alzaqebah, M. and S. Abdullah, Hybrid bee colony optimization for examination timetabling problems. Computers & Operations Research, 2015. 54: p. 142-154.

[14]. Bolaji, A.L.a., et al., University course timetabling using hybridized artificial bee colony with hill climbing optimizer. Journal of Computational Science, 2014. 5(5): p. 809-818.

[15]. Yan, H. and S.-N. Yu, A Multiple-Neighborhoods-Based Simulated Annealing Algorithm for Timetable Problem, in Grid and Cooperative Computing, M. Li, et al., Editors. 2004, Springer Berlin Heidelberg. p. 474-481.

[16]. Tuga, M., R. Berretta, and A. Mendes. A hybrid simulated annealing with kempe chain neighborhood for the university timetabling problem. in Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on. 2007. IEEE.

[17]. Ceschia, S., L. Di Gaspero, and A. Schaerf, Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Computers & Operations Research, 2012. 39(7): p. 1615-1624.

[18]. Gunawan, A., K.M. Ng, and K.L. Poh, A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem. Computers & Operations Research, 2012. 39(12): p. 3074-3088.

[19]. Burke, E.K., G. Kendall, and E. Soubeiga, A Tabu-Search Hyperheuristic for Timetabling and Rostering. Journal of Heuristics, 2003. 9(6): p. 451-470.

[20]. Arntzen, H. and A. Løkketangen, A Tabu Search Heuristic for a University Timetabling Problem, in Metaheuristics: Progress as Real Problem Solvers, T. Ibaraki, K. Nonobe, and M. Yagiura, Editors. 2005, Springer US. p. 65-85.

[21]. Karaboga, D., An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.

[22]. Awadallah, M.A., A.L.a. Bolaji, and M.A. Al-Betar, A hybrid artificial bee colony for a nurse rostering problem. Applied Soft Computing, 2015. 35: p. 726-739.

[23]. Das, P., et al. Arduino based multi-robot stick carrying by Artificial Bee Colony optimization algorithm. in Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on. 2015.

[24]. Abuhamdah, A., et al., Population based local search for university course timetabling problems. Applied intelligence, 2014. 40(1): p. 44-53.

[25]. Cacchiani, V., et al., A new lower bound for curriculum-based course timetabling. Computers & Operations Research, 2013. 40(10): p. 2466-2477.

[26]. Elloumi, A., et al., The classroom assignment problem: Complexity, size reduction and heuristics. Applied Soft Computing, 2014. 14: p. 677-686.

[27]. Chiarandini, M., et al., An effective hybrid algorithm for university course timetabling. Journal of Scheduling, 2006. 9(5): p. 403-432.

[28]. Al-Betar, M.A., A.T. Khader, and M. Zaman, University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2012. 42(5): p. 664-681.

[29]. Imanian, N., M.E. Shiri, and P. Moradi, Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems. Engineering Applications of Artificial Intelligence, 2014. 36: p. 148-163.

[30]. Karaboga, D. and B. Akay, A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation, 2009. 214(1): p. 108-132.

[31]. Ayob, M. and G. Jaradat. Hybrid ant colony systems for course timetabling problems. in Data Mining and Optimization, 2009. DMO'09. 2nd Conference on. 2009. IEEE.

[32]. Landa-Silva, D. and J. Obit, Evolutionary Non-linear Great Deluge for University Course Timetabling, in Hybrid Artificial Intelligence Systems, E. Corchado, et al., Editors. 2009, Springer Berlin Heidelberg. p. 269-276.