Solving Curriculum-Based Course Timetabling Problem with Artificial Bee Colony Algorithm

Danang Junaedi
Informatic Engineering
Universitas Widyatama
Bandung,Indonesia
danangjunaedi@gmail.com

Nur Ulfa Maulidevi
Informatic Engineering
Institute Techology of Bandung (ITB)
Bandung, Indonesia
ulfa@stei.itb.ac.id

***Abstract*— There is a trend in the scientific community to create and solve optimization models for complex problems based on the life in nature, one of which is Artificial Bee Colony Algorithm (Algorithm ABC). This paper discussed the ABC Algorithm to solve CBT (Curriculum-Based Course Timetabling) problems in the ITC-2007 (International timetabling Competition - 2007). The composition of the ABC algorithm to solve the problems is demonstrated in this paper. The proposed ABC algorithm consisting of four phases: initialization, Employed Bee phase, Onlooker Bee phase and phase Scout Bee. Initialization phase generating solution candidates which subjects were randomly placed in the slot and the space available. In Employed bee phase and Onlooker Bee phase, a search of existing slots and room around the subjects that violate the constraint with the aim of minimizing constraint violation is conducted. Based on the experimental results show that the ABC algorithm can solve problems of CBT on the ITC-2007 with fewer number of soft constraint violations compared to the results of Asaju et. Al, who proposed similar approach to solve CBT problems in the ICT-2007. However, the proposed ABC Algorithm is still not able to achieve results comparable with the best approach reported in the ITC-2007.**

Keywords- Artificial Bee Colony Algorithm, Curriculum-Based Course Time Tabling, International Timetabling Competition – 2007

I. INTRODUCTION

Timetable is a list of activities based on time, place and people of a set of activities. Timetable should be able to satisfy several things customized to the needs of implementing these activities. Time implementation of an activity must be created in such a way that no more than one activity at the same time [14].

Scheduling problem is a very complex problem. Viewed from the perspective of computer science formulations, most scheduling problems belong to the NP-complete problems, which mean no polynomial time algorithm can be used to solve them. Some cases of scheduling problems are scheduling in the world of education, sports, staff, transportation and other [8].

One aspect that determines the quality of a scheduling problem solver is the number of constraints that have violated. The less constraints are violated, the better the quality of the scheduling algorithm, although other factors are still need to be considered as well [11].

In curriculum-based course timetabling (CBT) problem, a set of courses should be placed in time slots and spaces that meet a certain set of constraints. A scheduling is expressed as a feasible schedule if all hard constraints are not violated. The objective of the curriculum-based course timetabling is to minimize the number of soft constraints that have been violated in a feasible schedule. CBT formulation issues that will be used derived from timetabling International Competition (ITC-2007). The main purpose of this competition is to reduce the gap between research and real-world cases of scheduling [13].

There is a trend in the scientific community to create and solve optimization models for complex problems based on the life in nature. This is caused by inefficiency of classical optimization algorithms in solving combinatorial problems and / or non-linear large-scale problems. There are many algorithms that are inspired from nature, such as genetic algorithms, simulated annealing and taboo search. And it has been proved that the algorithm produces better solutions than the classical algorithm [2].

Artificial Bee Colony (ABC) is a new member in the swarm intelligence that mimics the habits of honey bees to find food. Honeybees use some mechanism such as the waggle dance to optimize the location of food sources and find new sources. This caused it to be a good candidate to create new intelligent search algorithms [1].

Baykasoĝlu Adil et al. (2007) implemented artificial bee colony as a model for solutions relating to scheduling problem mentioned in Baykasoĝlu Adil et al. (2007). The paper adapted artificial bee colony algorithm to solve the problems partitioning/scheduling integrated in a code signing. The approach offered has been tested in a problem and the results were compared with genetic algorithms. The result is that the model generates better solutions than genetic algorithms, especially in terms of execution time and solution quality [1]. Nasser et. al. also adapting one model artificial bee colony algorithm to solve the exams scheduling problems. The model produces solutions that tend to be better than similar research [6]. Asaju et. al (2011) used artificial bee colony algorithm to solve the problem of curriculum-based course timetabling The algorithm produced very good results, though they were not comparatively better than those previously reported in the literature [14].

978-0-7695-4618-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ICI.2011.28

Based on the above explanation, we are interested to prove that artificial bee colony algorithm can be used to solve the problem of curriculum-based course timetabling (CBT) in this case minimizing the number of soft constraint violation and generate better solutions than the solutions produced by Asaju et. al[15].

We begin by identifying our problem, which is: first, what are the basic criteria that must be used in the university timetabling system. Second, how to model the university timetabling system by using this approach. And the third question is whether this approach can minimize the number of soft constraint violations

II. THE RESEARCH

A. The Research Handicap

In this research we bound to covered:

- Scheduling is applied in the case of Curriculum-Based Course timetabling (CBT) in the ITC (International timetabling Competition) 2007.
- We use the instance data with the format file.ctt at ITC (International timetabling Competition) 2007 obtained from the url http://tabu.diegm.uniud.it/ctt/index.php?page=instances.
- Constraint is used in accordance with the provisions of the ITC (International timetabling Competition) 2007

B. The Problem Characterization

Problems on the CBT (Curriculum-Based Course timetabling) in the ITC-2007 (International timetabling Competition - 2007) was to schedule all lectures from a collection of courses in the timetable per week, each lecture of a course should be placed in room and during the period of time in accordance with constraints that have been determined. In this issue, all the hard constraints must be fulfilled and minimize the number of soft constraint. A feasible schedule is a schedule in which all lectures are plotted on a certain period of time and room and satisfy all hard constraints. There are four hard constraints H1 to H4 and four soft constraints S1 to S4 as follows:

- H1. Lectures: All lectures of a course must be scheduled in distinct periods.
- H2 Room Occupancy: Two lectures cannot be assigned in the same room at the same time
- H3. Conflict: All lectures belonging to the same curriculum or taught by the same teacher must be scheduled in distinct periods
- H4. Availability: If a teacher is not available to give a lecture at a certain time, then the lecture has to be scheduled in another period
- S1. Room capacity: For every lecture, the number of students attending the lecture should be less than or equal to the number of classroom seats
- S2. Room Stability: All lectures of a course should take place in the same classroom. If not possible then it should minimize the number of rooms used
- S3. Minimum Working Days: Lectures of a course should be spread over a specified number of distinct working days
- S4. Curriculum Compactness: Lectures belonging to the same curriculum should be scheduled in consecutive periods

Candidate solution consists of a matrix $X_{p\times m}$ where $x_{i,j}$ contains the name of the course that scheduled in period t_i and room r_j. If no courses are scheduled in the period t_i and room r_j, then $x_{i,j}$ = -1. With this representation can be guaranteed no more than one course is scheduled in one room at any period, so that automatically hard constraint H2 is always fulfilled

The objective of this problem is to obtain a solution X that fisible (do not violate the hard constraint H1-H4) by minimizing the violations of soft constraints (S1-S4)

III. ARTIFICIAL BEE COLONY

Artificial Bee Colony (ABC) is an algorithm introduced by Dervis Karaboga in 2005, motivated by the intelligent behavior of honey bees. The algorithm is as simple as algorithms Particle Swarm Optimization (PSO) and Differential Evolution algorithm (DE), and only using the usual control parameters such as colony size and the maximum number of cycles. In the ABC system, bees flying around in a multi-dimensional search space, and some bees (worker bees and the onlooker) preferred food source depends on the experience and their partners, and then determine its position. Some fly and randomly selecting a food source without using any kind of experience. If the amount of nectar from a source more than the previous source (based on the bee's memory), then the new position will be stored in memory overwrite the old location. ABC system combines local search methods, carried out by workers and onlooker bees, with a global search method, managed by the onlookers and the Scouts, based on the balance and the exploration process [5]

Figure 1 is an illustration of the behavior of bees in search of food sources. Artificial bee algorithm shown in Algorithm 1.

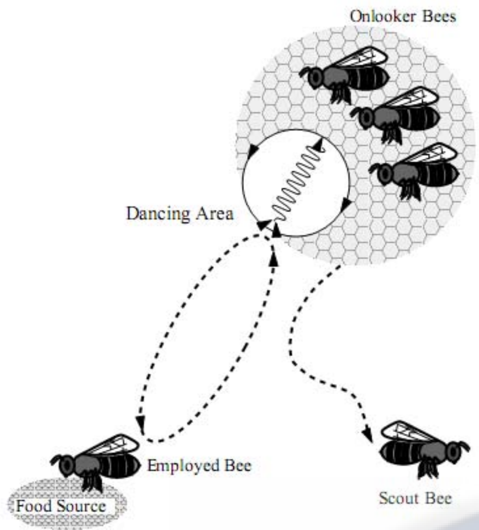

Figure 1. Basic elements of foraging behavior[7]

```
Begin
  Initialize the solution population x_i,i=1,…,SN
  Evaluate Population
  Cycle=1
  Repeat
    Generate new solution v_i for the employed bee using
                v_i,j = x_i,j + Φ_ij(x_i,j - x_k,j)  (1)
    Keep the best solution between current and candidate
    Select the visited solution for onlooker bees by their
    fitness
    Generate new solution v_i for the onlooker bee using (1)
    and evaluate them
    Keep the best solution between current and candidate
    Determine if exist an abandoned food source and replace
    it using scout bee
    Save in memory the best solution so far
    Cycle=Cycle + 1
  Until Cycle = MCN
End
```

Algorithm 1 Artificial Bee Colony [7]

IV. PROPOSED ARTIFICIAL BEE COLONY ALGORITHM

In this issue, all the hard constraints must be fulfilled and minimize the number of soft constraint. Feasible timetable is a timetable in which all lectures are plotted on a certain period of time and room and satisfy all hard constraints. There are four hard constraints H1 to H4 and four soft constraints S1 to S4 as follows:

- H1. Lectures. $\forall C_{IC} \in C$

$$\sum_{iR,iS} sl_{iC}\left(MK_{iR,iS}\right) = l_{iC} \qquad (2)$$

$$sl_{iC}\left(MK_{iR,iS}\right) = \begin{cases} 1 & if \quad MK_{iR,iS} = C_{iC} \\ 0 & if \quad MK_{iR,iS} \neq C_{iC} \end{cases}$$

- H2. Room Occupancy. $\forall MK_{iR1,iS}, MK_{iR2,iS} \in MK, MK_{iR1,iS} = c_u, MK_{iR2,iS} = c_v$, $\left(Sc_u = Sc_v\right) \wedge \left(Rc_u \neq Rc_v\right)$ (3)
- H3. Conflict. $\forall MK_{iR1,iS}, MK_{iR2,iS} \in MK, MK_{iR1,iS} = c_u, MK_{iR2,iS} = c_v$, $\left(\forall CR_{iCR}, c_u \notin CR_{iCR} \vee c_v \notin CR_{iCR}\right) \wedge \left(Sc_u \neq Sc_v\right)$ (4)
- H4. Availability. $\forall MK_{iR,iS} \in MK, MK_{iR,iS} = C_k$, $una_{k,iS} = 0$ (5)
- S1. Room capacity. $\forall MK_{iR,iS} \in MK, MK_{iR,iS} = C_k$,

$$f_{rc}\left(MK_{iR,iS}\right) = \begin{cases} \alpha_1\left(std_k - cap_j\right) & if \quad std_k > cap_j \\ 0 & if \quad std_k \leq cap_j \end{cases} \qquad (6)$$

- S2. Room Stability. $\forall C_{iC} \in C$,

$$f_{rs}\left(C_{iC}\right) = \alpha_2\left(NRuang_{iC}\left(X\right) - 1\right) \qquad (7)$$

- S3. Minimum Working Days. $\forall C_{iC} \in C$,

$$f_{md}\left(C_{iC}\right) = \begin{cases} \alpha_3\left(mwd_{iC} - NHari_{iC}\left(X\right)\right) & if \quad NHari_{iC}\left(X\right) < mwd_{iC} \\ 0 & if \quad NHari_{iC}\left(X\right) \geq mwd_{iC} \end{cases} \qquad (8)$$

- S4. Curriculum Compactness. $\forall MK_{iR,iS} \in MK, MK_{iR,iS} = C_k$,

$$f_{cc}\left(MK_{iR,iS}\right) = \alpha_4\left(\sum_{cr_{iCR} \in CR} \mathrm{CRCourse}_{k,iCR} \cdot iso_{iCR,iS}\left(X\right)\right) \qquad (9)$$

where

$$\mathrm{CRCourse}_{k,iCR} = \begin{cases} 1 & if \quad c_k \in CR_{iCR} \\ 0 & sebaliknya \end{cases}$$

$$iso_{iCR,iS}\left(X\right) = \begin{cases} 1 & if\left(\left(iS / JDays\right) = \left(\left(iS - 1\right) / JDays\right) \wedge cra_{iCR,iS-1}\left(X\right) = 0\right) \wedge \\ & \left(\left(iS / JDays\right) = \left(\left(iS + 1\right) / JDays\right) \wedge cra_{iCR,iS+1}\left(X\right) = 0\right) \\ 1 & if\left(\left(iS / JDays\right) = \left(\left(iS - 1\right) / JDays\right) \wedge cra_{iCR,iS-1}\left(X\right) = 0\right) \\ 1 & if\left(\left(iS / JDays\right) = \left(\left(iS + 1\right) / JDays\right) \wedge cra_{iCR,iS+1}\left(X\right) = 0\right) \\ 0 & others \end{cases}$$

The purpose of this problem is to obtain that feasible timetable (does not violate the hard constraint) and minimizing the number of soft constraint violations of a solution. The number of soft constraint violations for a solution computed using the following equation:

$$f\left(X\right) = \left(\sum_{x_{i,j} \in X} f_{rc}\left(x_{i,j}\right) + \sum_{c_i} f_{rs}\left(c_i\right) + \sum_{c_i \in C} f_{md}\left(c_i\right) + \sum_{x_{i,j} \in X} f_{cc}\left(x_{i,j}\right)\right)$$

(10)

where H1 ~ H4 = 0, and α1 =1,α2 =1,α3 =5,α4 =2 (based on reference from the ITC-2007).

For the solution used a representation in matrix form

$$MK_{iR,iS} \quad (11)$$

where $MK_{iR,iS} \in C, iR \in R, iS \in S$

MK contains iC (index course C_{iC}) that is scheduled in the period iS and room iR. If no courses that scheduled in the period iS and room iR iR, then $MK_{iR,iS} = -1$, $C = \{c_1, c_2, ..., c_n\}$. A period $S_a \in Slot$ a pair of days and time slots, as well as the period iS devided into d days in a week and daily time slots h (S = $<d_i,h_j>$), d={$d_1,d_2,\ldots,d_n$}.where d_i is i-th day and n maximum numbers of day in a week, h={$h_1,h_2,\ldots,h_m$} where h_i is daily time slot and m the maximum number of slots in one day. R = {$r_1,r_2,...,r_n$} is the room that used .

ABC algorithm proposed by Asaju et. al using two approaches: first, Saturation Degree (SD) was used to Ensure a feasible solution and secondly, Artificial Bee Colony Algorithm was used to further improve the results obtained[15]. The difference between the proposed ABC algorithm and the algorithm version Asaju et. al is Artificial Bee Colony algorithms is simultaneously used to generate feasible solutions and to minimize the number of soft constraint violations. The proposed Artificial Bee Colony Algorithm for CBT problem is as follows:

0. Initialization parameters and read the data instance NP,FoodNumber, jEB, jOB, jSB, MCN
1. Initialize population of solutions X_i, i=1..FoodNumber
2. Evaluate the population with the equation (10)
3. cycle = 1
4. REPEAT

   //Fase *Employed Bees*$_i$, i=1..jEB
5. Swap(jTetangga1),Evaluasi
6. IF jSoftConstraint(Swap(jTetangga)) < jSoftConstraint (*Employed*) THEN
   Employed= Swap(jTetangga)
7. Swap(jTetangga2,2), Evaluasi
8. IF jSoftConstraint (Swap(jTetangga,2))< jSoftConstraint (*Employed*) THEN
   Employed= Swap(jTetangga,2)
9. Calculate the probability P_i, i=1…FoodNumber with the equation (12)

   $$P_i = \frac{fit_i}{\sum_i fit_i} \quad (12)$$

   where fit calculated using the equation (13)

   $$fit_i = \frac{1}{1+f(X)} \quad (13)$$

   And f(X) calculated with equation (10)

   //Fase *Onlooker Bee*$_i$, i=1…jOB
10. Send P_i*jOB Onlooker Bee to *Employed Bees$_i$*
11. Swap(jTetangga3,MaxMove), Evaluasi
12. IF best(jSoftConstraint (Onlooker))< jSoftConstraint (*Employed*) THEN
    Employed= Onlooker
    //Best Solution
13. IF best$_{cycle-1}$(jSoftConstraint (*Employed$_h$*))<best(jSoftConstraint (*Employed$_i$*)) THEN
    Best$_{cycle}$= *Employed$_h$*
    ELSE
    Best$_{cycle}$= *Employed$_i$*
    //Fase *Scout Bee*$_i$, i=1…jSB
14. Initialization *Scout Bee*, evaluate
15. IF worst(jSoftConstraint (*Employed$_j$*))> jSoftConstraint (Scout$_i$) THEN
    Employed$_j$=*Scout$_i$*
16. cycle = cycle + 1
17. Until cycle = MCN

Algorithm 2 Artificial Bee Colony for CBT Problems

Examples of the swap process in Algorithm 2 above can be seen in Figure 2.

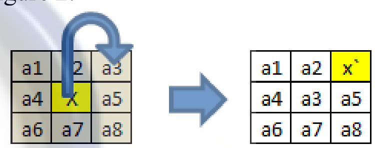

Figure 2. Illustration for Swap Process

V. Experiment

To evaluate the performance of the algorithm we conducted experiments on 21 data instances with different parameters. Each data instance is run 10 times using different parameters. All computing processes of each of the experiments carried out without any special settings to the parameters used (in other words a constant value). So it did not rule out the best solution for each instance generated by using different parameters.

Based on Algorithm 2 we developed an application using C++ programming language with compiler MinGW 5.1.6, whereas experiments performed on 3 computers with the specifications: (1) Processor Intel i3 2.13GHz, Memory 2 GB, Operating System: Windows 7 32 bit, (2) Processor Intel Pentium 4 2.40GHz, Memory 512 MB, Operating System: Windows XP., (3) Processor Intel Pentium 4 2.40 GHz, Memory 256 MB, Operating System: Windows XP

We conducted five experiments consists of:

- The first experiment performed to determine the value jTetangga and MaxMove in Algorithm 2 to produce the best solution in terms of average number of soft constraint violations. Value for each parameter used for this experiment are NP=10, jEB=50%NP, jOB=50%NP, jSB=10%NP, MCN=5.
- The second experiment conducted to determine the effect of population on the quality of the solution. Value for each parameter used for this experiment are NP={10, 20,60}, jEB=50%NP, jOB=50%NP, jSB=10%NP, MCN=5.

- The third experiment performed to determine the effect of the number of cycles on the quality of the solution. Value for each parameter used for this experiment are NP={10, 20,50}, jEB=50%NP, jOB=50%NP, jSB=10%NP, MCN={10, 20, 30}.
- The fourth experiment conducted to determine the effect of the number of onlooker bees on the quality of the solution. Value for each parameter used for this experiment are NP=60, jEB=50%NP, jOB=NP, jSB=10%NP, MCN=6.
- The fifth experiment is performed to determine the parameter value that generates solutions that approach the winner of the ITC-2007. Value for each parameter used for this experiment are NP=60, jEB=50%NP, jOB=NP, jSB=10%NP. For MCN parameter value is determined by a comparison the average value of four experiments with the highest value of the 6 finalists ITC-2007 MCN multiplied by 60.

The conclusions from the experiments are as follows:

- The quality of the solution influenced the following parameters:
  a. Number of population, greater number of population, the better the resulting solution. Probably caused by the number of candidate solutions for more, so alternative solutions are generated more.
  b. Number of Onlooker Bee, greater number of Onlooker Bee the less the number of soft constraint violations. In phase onlooker bee, previously optimized candidate solutions, so the more the number of onlooker, the more often a candidate solution is optimized.
  c. Number of cycles, greater number of cycles the number of soft constraint violations tend to decrease. At each cycle solutions with quality candidates sought a better solution. The more cycles that performed it tend to get a better solution
- The lower the soft constraint violations in the initialization phase will tend to produce better solutions (the number of soft constraint violations is lower). This occurs because the optimization processes on the candidate solution with a number of soft constraint violations are low tend to produce solutions with better quality.
- The quality of the solution also influenced the amount of space, constraints and events to be held. The fewer the number of rooms are available as well as the greater number of constraints and events to be held (including comp05 and comp12), the more soft constraint is violated.
- In the phase of scout bee, the worst employed bee almost never being overwritten with a new solution candidate's of scout bee, so the solution tends to converge on a specific candidate solution (employed bee).

The experimental results can be seen in the TABLE I below:

TABLE I. THE ABC ALGORITHM EXPERIMENTAL RESULTS

	Proposed ABC Algorithm		ABC Algorithm by Asaju et.al. [15]		Best in ITC-2007 [13]
Instance	Best	Worst	Best	Worst	
Comp01	23	29	-	-	5
Comp02	190	254	312	391	50
Comp03	171	211	292	357	71
Comp04	132	145	193	224	35
Comp05	1483	2405	-	-	309
Comp06	237	254	336	392	48
Comp07	259	281	324	410	20
Comp08	154	176	218	246	40
Comp09	190	221	302	339	104
Comp10	210	225	274	371	16
Comp11	18	24	293	359	0
Comp12	583	663	-	-	333
Comp13	156	172	-	-	66
Comp14	165	186	236	295	57
Comp15	193	215	284	335	71
Comp16	215	236	281	330	3
Comp17	206	241	331	376	83
Comp18	122	146	196	217	68
Comp19	205	231	304	314	62
Comp20	263	291	372	457	27
Comp21	233	276	-	-	103

Based on the results of the comparison in Table 1 above the proposed ABC algorithm can solve problems of CBT on the ITC-2007 with fewer number of soft constraint violations compared to the results of Asaju et. al. However, the proposed ABC Algorithm is still not able to achieve results comparable with the best approach reported in the ITC-2007.

VI. CONCLUSIONS

Upon developing ABC algorithm, we concluded:

- In the initialization phase of the ABC algorithm in this paper, subjects were randomly placed in the slot so it would produce a solution with a number of soft constraint violations that tend to be higher.
- Based on the experimental results, the diversity of candidate solutions affect the quality of the resulting solution so that the greater number of population, the better solution quality is yielded by onlooker.
- Based on observations of experimental results, the resulting schedule has been feasible but there are still a lot of soft constraint violations. One possible reason is that for a certain solution candidate, the cycles tend to have convergent in a particular phase,

so the search mechanism for a solution needs to be considered as the addition to overcome the problem.

- As the results have shown, the algorithm produces fewer number of soft constraint violations compared to the results of Asaju et. al.
- ABC algorithm is used by Dervis Karaboga to solve numerical optimization problems with good results. Based on that and based on the experiment result, the ABC on numerical problem can produce a better solution quality than symbolic problem such as CBT.

For further development we suggest that in the initialization phase, courses placement in a particular slot should not be made completely random, but should be a mechanism so that the number of constraint violations resulting in a candidate solution can be kept to a minimum. In Onlooker Bee phase should be considered a mechanism to avoid premature convergence, for example, if a solution has a premature indication of convergent exchange position subjects performed randomly in the hope of obtaining a solution with better quality. And the overwriting mechanism of the worst solution in the phase scout bee should be made, to increase diversity candidate solutions that expected to improve the quality of the final solution

REFERENCES

[1] Adil Baykasoğlu, Lale Özbakır and Pınar Tapkan, ”*Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem”*, Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria, doi: 10.4249/scholarpedia.6915

[2] Barry McCollum, “*University Timetabling : Bridging the Gap between Research and Practice”*, PATAT,2006, ISBN: , 3-540-77344-4 978-3-540-77344-3, doi:10.1.1.61.7820

[3] B. Mccollum , P. Mcmullan , B. Paechter , R. Lewis , A. Schaerf , L. Di Gaspero , A. J. Parkes , R. Qu , E. Burke, “*Setting the Research Agenda in Automated Timetabling: The Second International Time-tabling Competition”*, ITC-2007 Technical Report, 29 August 2008, doi=10.1.1.153.4016

[4] Chin Soon Chong, Malcolm Yoke Hean Low, Appa Iyer Sivakumar, and Kheng Leng Gay, “*A Bee Colony Optimization Algorithm To Job Shop Scheduling”*, Proceedings of the 2006 Winter Simulation Conference, 2006, ISBN: 1424405017, Pages: 1954-1961, doi=10.1.1.109.3897

[5] Dervis Karaboga dan Bahriye Akay, “*A comparative study of Artificial Bee Colony algorithm”*, Jurnal Applied Mathematics and Computation 214 (2009) ISSN: 00963003, pp.108–132, 2009, • DOI: 10.1016/j.amc.2009.03.090

[6] Nasser R. Sabar, Masri Ayob, Graham Kendall, “*Solving Examination Timetabling Problems using Honey-bee Mating Optimization (ETP-HBMO)*”, Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009), Dublin, Ireland, 10-12 August 2009

[7] Omar Cetina Domìnguez, “*An adaptation of the scout bee behavior in the Artificial Bee Colony algorithm to solve constrained optimization problems”*, Master of Computer Science, Laboratorio Nacional de Inform´atica Avanzada A.C.,2009

[8] Rhydian Marc Rhys Lewis, “*Metaheuristics for University Course Timetabling”*, Napier University, 2006, doi= 10.1.1.62.1475

[9] Saif Mahmood Saab, et al. 2010. ”*Developing Optimization Algorithm Using Artificial Bee Colony System*”. http://www.ubicc.org/files/pdf/Developing_Optimization_Algorithms_Using_Artificial_Bee_Colony_System.doc_391.pdf, 5 May 2010

[10] Shoshana Hahn-Goldberg,”*Defining, Modeling, and Solving a Real University Course Timetabling Problem*”, Masters thesis, University of Toronto, 2007

[11] J.D. Landa Silva, E.K. Burke, S. Petrovic,”*An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling*”, Automated Scheduling, Optimisation and Planning Research Group School of Computer Science and IT, University of Nottingham, UK, *Metaheuristic for Multiobjective Optimisation*, Xavier Gandibleux, Marc Sevaux, Kenneth Sorensen, Vincent T'kindt (eds.), Lecture Notes in Economics and Mathematical Systems, Vol. 535, pp. 91-129, Springer, 2004, doi=10.1.1.63.8888

[12] Tomáš Müller,”*Constraint-based Timetabling*”, Ph.D. Thesis, Faculty of Mathematics and PhysicsCharles University in Prague, 2005, doi=10.1.1.136.6504

[13] Zhipeng L and Jin-Kao Hao,”*Solving the Course Timetabling Problem with a Hybrid Heuristic Algorithm*”, AJMSA,2008, ISBN: 978-3-540-85775-4 doi:10.1007/978-3-540-85776-1_22

[14] Willemen, Robertus J., “*School timetable construction : algorithms and complexity”*, Technische Universiteit Eindhoven, 2002

[15] Asaju La’aro Bolaji, Ahamad Tajudin Khader, Mohammed Azmi Al-betar, Mohammed Awadallah,” Artificial Bee Colony Algorithm for Curriculum-Based Course Timetabling Problem”, The 5th International Conference on Information Technology (ICIT 2011), 2011