University Course Timetabling Model in Joint Courses Program to Minimize the Number of Unserved Requests

Purba Daru Kusuma[1]
Computer Engineering
Telkom University, Bandung, Indonesia

Abduh Sayid Albana[2]
Industrial Engineering
Institut Teknologi Telkom Surabaya, Surabaya, Indonesia

***Abstract*—This work proposes a novel course timetable model for the national joint courses program. In this model, the participants, both students and lecturers, come from different universities. It is different from most existing university course timetabling models where the environment is physical, and the system can dictate the timeslots and classrooms for the students and lecturers. The courses are delivered online in this model, so physical classrooms are no longer required, as was the case in most previous course timetabling studies. In this model, the matching process is conducted based on the assigned timeslots and the requested courses. The courses are elective rather than mandatory. Three metaheuristic methods are used to optimize this model: artificial bee colonies, cloud theory-based simulated annealing, and genetic algorithms. Due to the simulation process, the cloud theory-based simulated annealing performs best in minimizing the number of unserved requests. This method outperforms the two other metaheuristic methods, the genetic algorithm, and the artificial bee colony algorithm. According to the simulation results, when the number of students is low, the cloud theory-based simulated annealing has 91 percent fewer unserved requests than the genetic algorithm. When the number of students is large, this figure drops to 62%.**

***Keywords*—Course timetabling; joint course program; artificial bee colony; simulated annealing; genetic algorithm; online course**

I. Introduction

In January 2020, the Ministry of Education and Culture of Indonesia launched a national program called Merdeka Belajar Kampus Merdeka (MBKM). This program is conducted for undergraduate students in Indonesia. Through this program, students can take courses or earn credits outside of their study program. For example, these students can take several courses from other universities. Besides, students can also take courses which are not related to their study program. For example, a computer engineering student can take management or accounting courses from outside his university. This general program is then followed by several joint programs. In this joint program, several universities provide several courses together. Each university provides several courses so that external students can attend these courses. In some programs, students can select universities and the related courses (lecturers) explicitly. In other programs, students just select the courses without knowing the lecturers who provide the selected courses so that the students cannot choose the lecturers. As a national program, the provided courses are conducted online so that geographic barriers do not matter, for example, in rural and remote areas [1]. It also minimizes the cost due to providing the physical classrooms [1].

Despite the fact that this program gives benefits to students, especially in the flexibility, affordability, and accessibility aspects [1], there is a problem in arranging the courses. Students who follow this national program still need to attend the physical courses conducted at their own universities. The lecturers who participate in this program also teach at their own universities. Both students and lecturers have their own schedule. Based on that, schedule matching between lecturers and students in this national program becomes a critical issue.

Unfortunately, existing course timetabling models in many studies cannot be implemented directly to solve this problem. The main reason is that, in general, the existing course timetabling models were conducted for single department [2] or university [3,4]. The system has full authority to allocate the resources (lecturers, physical rooms, and timeslots) and manage the students. The system can dictate the timeslots for both students and lecturers. Besides, most course timetabling studies were conducted in a physical environment where the courses were conducted in physical rooms, so that the limited number of rooms became a constraint [5,6]. In this national joint course program, the courses are conducted online so that the physical rooms are not needed anymore. On the other hand, the system cannot dictate timeslots for both students and lecturers.

Based on this problem, this work aims to develop a course timetabling model that suits the circumstances of this national joint course program where the participants are students and lecturers from different universities. Both students and lecturers choose their available timeslots. A lecturer handles one specific course only. A student can take or request several courses provided by the program. The objective of this model is to minimize the number of unserved requests. In the context of a timetabling study, the number of unserved requests becomes the soft constraint.

Like the existing course timetabling studies where the models were optimized by using metaheuristic techniques, for example: genetic algorithm [7,8], simulated annealing [9,10], tabu search [2], and so on, this model is also optimized by using three metaheuristic methods: artificial bee colony algorithm, cloud theory based simulated annealing, and genetic algorithm. The artificial bee colony and simulated annealing are chosen due to their advantage in finding global optimization by avoiding local optimal traps [11]. The cloud theory-based simulated annealing is a derivative of the basic simulated annealing that gives a faster process [12].

The contributions to this work are as followed:

- This work proposes a novel course timetabling for a joint online course program where the students and the lecturers come from many universities.
- The proposed model arranges course timetables based on the available timeslots and requested courses.
- Its objective is to minimize unserved requests, which is rare in many existing course timetabling studies.

This proposed model can be used practically in many joint courses program. It can be applied not only in national scale program, as it is stated in the opening paragraph, but in smaller scale. For example, universities with same foundation or same area (district or province) can make a joint courses program. Besides, the joint courses program can be conducted for universities with same subjects, such as computer science, finance, law, and so on. The key is that the joint program is conducted in voluntary and online based approach so that this proposed model can be applied.

This paper is organized as follows. The background, research purpose, and the contribution are explained in section one. The theoretical aspects of course timetabling and several recent studies in course timetabling are reviewed in Section two. The proposed model is described in Section three. The simulation and result are presented in Section four. The findings are discussed in Section five. The conclusion of this work and the future research potential are explained in Section six.

II. Related Work

A timetable can be defined in several ways. Aziz and Aizam [13] defined a timetable as a table with all the data of events and the information about these events, such as time and place. Alghamdi, Alhakami, Alsubait, and Baz [14] defined a timetable as a table of various events or activities with their schedule. Zhu, Li, and Li [7] defined timetabling as a process of distributing activities among limited resources (place and time). The timetabling problem can be categorized as an NP-complete problem, so it is difficult to find the general optimal solution [7].

The educational timetable problem has become the most well-known among the timetable studies. An educational timetable problem is a combinatorial problem with the objective of assigning a certain number of didactic activities to a certain room within certain timeslots [3]. The entities are courses, instructors, rooms, and registered students [5]. Educational timetabling can be divided into three categories: course timetabling, school timetabling, and examination timetabling [7]. Course timetabling is assigning lecturers and courses to timeslots, rooms, and other facilities [7]. School timetabling involves assigning teachers or instructors to courses based on their specialization [7]. School timetabling can also be called curriculum-based course timetabling [7]. Examination timetabling is assigning exams into rooms and timeslots [7]. Several factors that affect the educational timetable design are the number of courses, the average number of lectures per day, the targeted free timeslots per day, and targeted off-days in a week [14].

There are two types of constraints in the timetabling problem: hard constraints and soft constraints. Hard constraints are constraints or rules that cannot be violated [15]. Soft constraints are constraints that can improve the performance of the timetable if they are not violated [15]. Several common hard constraints are as follows:

- A lecturer can only teach a course at a certain time [3,16].
- A student cannot attend more than one lecture at one time [5].
- The attendants cannot surpass the room's capacity [3,5].
- The timeslot must be conducted within a certain time window [3].

Meanwhile, there are several soft constraints used in several educational timetabling studies. These soft constraints are as follows:

- There is a minimum number of courses in a day for the students [5].
- There is a maximum number of courses in a day for the lecturers [5,6].
- Lecturers may have preferred teaching timeslots [16].
- Lecturers may have minimum working days [17].
- Lecturers may have preferred classrooms [6].
- The last timeslots of the day should be avoided [2].

There are many studies conducted on this educational timetabling problem. Each study is developed based on its specific circumstances, objectives, and methods. Most studies use computational methods, especially metaheuristic methods, to find the optimal solution. Table I shows the recent studies on the educational timetable problem with their objectives and methods. These studies were conducted from 2016 to 2021. They are presented chronologically.

There are several notes on this presented literature. First, most studies on the course timetabling problem were conducted in face-to-face interaction between students and lecturers, so the number of limited physical rooms becomes a constraint. Second, all these studies were conducted in a department or university so that the system could dictate the timeslot allocation.

Based on these notes, this work proposes a novel course timetabling model due to specific circumstances in the national joint courses program. These circumstances are not found in the existing course timetabling studies. First, this joint program is conducted online. Second, the system cannot dictate the timeslots for both students and lecturers. The matching process is conducted based on the available timeslots that are allocated by the students and lecturers. Third, the courses are open elective so that students can choose any course provided in the system and they can choose more than one course. Like the existing studies, this work will use several metaheuristic techniques to optimize the solution. These techniques will be compared to each other.

TABLE I. RECENT STUDIES IN EDUCATIONAL TIMETABLING PROBLEM

Authors	Objectives	Methods
[4]	improve resources utilization	genetic algorithm
[5]	minimize conflicts	genetic algorithm
[8]	minimize penalty	multi-objective genetic algorithm, hill climbing, simulated annealing
[17]	minimize total sum of penalty points	mixed integer linear programming (MILP), large neighborhood search
[10]	minimize clashes	genetic algorithm, simulated annealing
[6]	optimize resources	genetic algorithm
[19]	minimize the number of classrooms	linear programming
[2]	minimize students' maximum number of events per day, avoid the usage of the last timeslot of the day	tabu search, variable neighborhood search
[20]	improve accuracy	genetic algorithm, supervised learning (regression and classification)
[22]	maximize lecturers' presence time and education quality	three-stage heuristic algorithm
[3]	minimize idle time	genetic algorithm
[18]	reduce redundant workload, improve classroom seat utilization	genetic algorithm, fuzzy pattern algorithm
[21]	minimize soft constraints	integer linear programming (ILP), branch-and-bound algorithm

III. PROPOSED MODEL

This model consists of several entities. A student is a person who takes a course or several courses. A lecturer is a person who delivers courses to students. A course is a unit of teaching that is delivered by a lecturer, for example: artificial intelligence, machine learning, mobile programming, and so on. A class is a group of students that take the same course and are taught by the same lecturer with the same timeslot. A Request is a course requested by a student.

The system consists of a certain number of students and lecturers. The students come from any universities, and so do the lecturers. The relationships between students and lecturers are many to many. It means a student can be taught by more than one lecturer depending on the number of courses that this student takes. The number of students is greater than the number of lecturers. Meanwhile, a lecturer can teach many students. This relation is shown in Fig. 1.

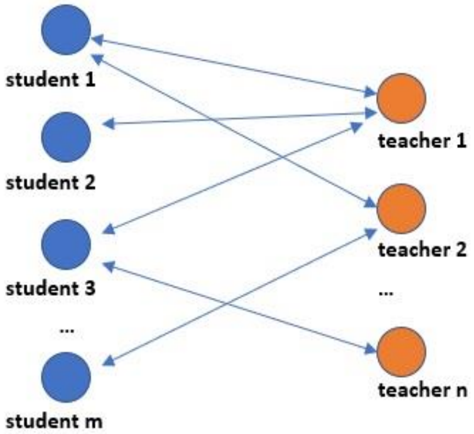

Fig. 1. Student-Lecturer Relationship.

In this system, a lecturer can only teach a course due to the expertise factor. Meanwhile, a student can take several courses. Both lecturers and students assign several available timeslots. These timeslots are week-based timeslots. It means there are a certain fixed number of timeslots in a week. The number of timeslots that are booked by a lecturer represents the number of classes that are taught by him or her. A student can take a course that is delivered by a lecturer as long as the course in this class is the same as the course that is taken by this student and the class timeslot is the same as the student's timeslot. The illustration of this relationship is shown in Fig. 2.

The explanation of Fig. 2 is as follows. There is a lecturer who teaches a course, for example, course A. He assigns two timeslots for this course so that he handles two classes, class 1 and class 2. Meanwhile, there are five students who want to take course A. Based on the matched timeslot, three students (student 1, student 2, and student 3) are assigned to class 1 and two students (student 4 and student 5) are assigned to class 2.

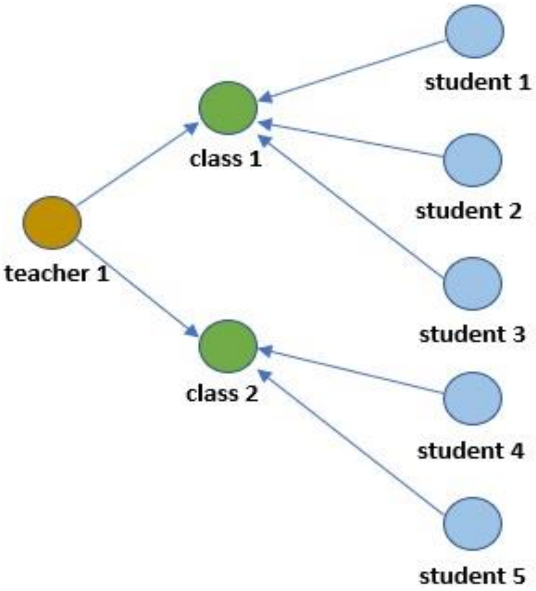

Fig. 2. Illustration of Classes.

This model is developed based on several hard constraints as follows:

- Students are not permitted to attend more than one class at a time [3].
- A lecturer is not permitted to teach more than one class at a time [5].
- A student cannot be assigned to a class where its timeslot is outside of his available timeslots [3].
- A student cannot be assigned to a class where the course is not his requested course.
- Students cannot request a course outside of the provided courses.
- A lecturer cannot teach a class where the timeslot is outside of his available timeslots [3].
- The number of class attendants cannot exceed the maximum capacity of the class [5].
- A class cannot be plotted outside of the available timeslots in a week.

This model is also developed by using several annotations as follows:

c	class
c_{sel}	selected class
C	set of classes
C_{pos}	set of possible classes
s	student
S	set of students
l	lecturer
L	set of lecturers
n	number of entities
o	course
O	set of courses
q	quantity (number of attendants)
q_{max}	maximum quantity
r	request
R	set of requests
v	status
t	timeslots
T	set of timeslots
T_{pos}	set of possible timeslots
u	unserved request
U	set of unserved requests
v_q	class availability status based on capacity
v_t	class availability status based on timeslot
v_o	class availability status based on course
v_{at}	student's availability status at certain timeslot

The objective of this model is to minimize the number of unserved requests. This objective is formalized by using (1) and (2). Equation (2) shows that the unserved request is a request where its status is 0. Meanwhile, if this request is served, its status will be 1. In this system, a request can be unserved because of several problems. First, there is a mismatch between the timeslot of the request and the timeslot of the available classes for the same course. Second, there is a class or several classes whose course is the same as the course of the request, and the timeslot of the class is also matched, but the number of attendants has reached the maximum capacity of this class.

$$min\big(n(U)\big) \tag{1}$$

$$u = r|v_r(r) = 0 \tag{2}$$

At the beginning, both the lecturers and the students make preparations. The lecturers assign their own timeslots or classes. Meanwhile, the students choose courses that they want to take and their available timeslots in a week. This lecturer's set up is formalized by using (3) and (7). Meanwhile, the student's set up is formalized by using (8) and (10).

$$C(l) = \{c|o(l) = o(c) \wedge c \in C\} \tag{3}$$

$$n\big(C(l)\big) = n\big(T(l)\big) \tag{4}$$

$$T(l) = \{t|t \in T\} \tag{5}$$

$$t(c_i) \neq \forall t(c)|c \in C(l) \wedge c_i \in C(l) \wedge c \neq c_l \tag{6}$$

$$n(C) = \textstyle\sum_{\forall l} n\big(C(l)\big) \tag{7}$$

The explanations for (3) to (7) are as follows. Equation (3) declares that all classes that are handled by a lecturer must have the same course as the lecturer. Equation (4) shows that the number of classes handled by a lecturer is equal to this lecturer's number of timeslots. Equation (5) shows that the lecturer's timeslots must be within the allocated timeslots in a week. The hard constraint where a lecturer can only visit one class in a timeslot is formalized in (6). Equation (7) shows that the number of classes is equal to the accumulation of the number of classes of all lecturers.

$$R(s) = \{r|o(r) \in O\} \tag{8}$$

$$T(s) = \{t|t \in T\} \tag{9}$$

$$n(R) = \textstyle\sum_{\forall s} n\big(R(s)\big) \tag{10}$$

The explanation of (8) to (10) is as follows. Equation (8) states that student can request courses within the provided courses. Equation (9) states that the students' timeslots must be within the provided timeslots in a week. Equation (10) shows that the total number of requests is the accumulation of all students' requests.

After setup, the next process is the matching process. This process is conducted iteratively from the first request to the last request. The matching order is shuffled so that the first request is not prioritized rather than the last process. Each request consists of two attributes: the student and the course. This matching process is formalized by using (11) to (17).

$$c_{sel}(r) = rand(C_{pos}(r)) \tag{11}$$

$$C_{pos}(r) = \big\{c|v_q(c) = 1 \wedge v_t(r,c) = 1 \wedge v_o(r,c) = 1\big\} \tag{12}$$

$$v_q(c) = \begin{cases} 1, q_c(c) < q_{max} \\ \quad 0, else \end{cases} \tag{13}$$

$$v_t(r,c) = \begin{cases} 1, \exists t_{pos}\big(s(r)\big) = t(c) \\ \quad 0, else \end{cases} \tag{14}$$

$$T_{pos}(s(r)) = \{t | t \in T(s) \wedge v_{at}(t,s) = 1\} \tag{15}$$

$$v_{at}(t,s) = \begin{cases} 1, \forall t(s), \nexists t(r(s)) = t \\ 0, else \end{cases} \tag{16}$$

$$v_o(r,c) = \begin{cases} 1, o(r) = o(c) \\ 0, else \end{cases} \tag{17}$$

The explanation of (11) to (17) is as follows. Equation (11) shows that the class is selected randomly among the possible classes for the request. Equation (12) states that the possible class must meet three aspects: capacity, time, and course. Equation (13) shows that the class is available if its current number of attendants is still less than its maximum capacity. Equation (14) shows that the class meets the time aspect if there exists a possible student's timeslot that is the same as the timeslot of this class. Equation (15) shows that the set of possible timeslots of the student consists of the student's timeslots that are still available. Equation (16) shows that the student's timeslot is available if it has not been occupied by any other student's request. Equation (17) shows that the class meets the course aspect if the course of the class is the same as the course of the request.

In this work, we compare three metaheuristic algorithms: the genetic algorithm (GA) [18], cloud theory-based simulated annealing algorithm (CSA) [12], and artificial bee colony algorithm (ABC) [23]. As metaheuristic algorithms, they consist of a stochastic approach, especially during the initialization process and the improvement process. In the genetic algorithm method, the half best solution becomes the new generations during the reproduction process.

All these algorithms are population-based algorithms. A population consists of individuals or solutions. An individual consists of an array of requests. Each element consists of attributes: request, student, and class.

During the improvement process, the pairwise interchange is conducted by selecting a served request, finding a new class, assigning this request to the new class, and then allocating the abandoned seat to another unserved request. This pairwise interchange process is formalized by the use of algorithm 1.

Algorithm 1: pairwise interchange

```
$r_{sel1}$ = rand ($R$, $v_r(r)$ = 1)
$c_{sel1}$ = rand ($C$, $r_{sel1}$)
**if** found ($c_{sel1}$) **then**
  assign ($r_{sel1}$, $c_{sel1}$)
  $r_{sel2}$ = rand (U)
  **if** found ($r_{sel2}$) **then**
    $c_{sel2}$ = rand ($C$, $r_{sel2}$)
    **if** found ($c_{sel2}$) **then**
      assign ($r_{sel2}$, $c_{sel2}$)
    **end if**
  **end if**
**end if**
```

IV. Simulation and Result

This proposed model is then implemented into the course timetabling simulation. As mentioned before, there are three metaheuristic techniques that are used to optimize the model: the genetic algorithm (GA), cloud theory-based simulated annealing algorithm (CSA), and artificial bee colony algorithm (ABC). The objective function applied in these algorithms is to minimize the number of unserved requests. The reason is that this objective function is related to the objective of this work and proposed model.

The technical parameters used in these metaheuristic algorithms are as follows. In the GA, the population size is 10 individuals and the maximum number iterations is 100 iterations. In the CSA, the population size is 5 solutions, the initial temperature is 100, the termination temperature 50, and the number of iterations is 10. In the ABC, the population size is 10 bees, the maximum number of iterations is 50 iterations, and the limit is 30.

The scenario used in this simulation is as follows. In the beginning, a certain number of students and lecturers are generated. A lecturer teaches only one course. A student can request several courses. Both students and lecturers select their available timeslots. The number of timeslots that are chosen by the lecturer represents the number of classes that he will handle. The students will be allocated based on the timeslots that they have chosen. The simulation then runs based on this initial setting. During the simulation process, students will be matched with the available classes based on their selected course timeslots. At the end of the simulation, certain requests may be unserved due to a mismatch.

There are two simulations conducted in this work. The observed parameter is the number of unserved requests. This parameter also becomes the fitness function of these three metaheuristic techniques (minimizing the number of unserved requests). There are several adjusted parameters that are set as default. There are five courses in the system. The number of lecturers is 20. The maximum capacity of each class is 40 attendants. There are 20 timeslots that can be chosen by both students and lecturers. The number of courses that are requested by a student is generated randomly and follows a normal distribution. The average number of requests is 3 courses. The number of timeslots that are chosen by both lecturers and students is also generated randomly and it follows normal distribution.

The first simulation is conducted to observe the relation between the number of students with the number of unserved requests. The number of students ranges from 400 to 600 students, with a step size of 20 students. The average number of chosen timeslots is 3 timeslots. The result is shown in Fig. 3.

In Fig. 3, it is shown that the number of unserved requests increases due to the increase in the number of students. This trend occurs in all methods. The rationale for this condition is that demand is increasing, whereas supply is still the same, so the scarcity is also increasing. Compared among methods, cloud theory-based simulated annealing performs as the best model in creating the lowest number of unserved requests. On the other hand, the genetic algorithm performs as the worst method. The artificial bee colony performs moderately. When the number of students is low (400), the simulated annealing algorithm generates 91% fewer unserved requests than the genetic algorithm. When the number of students is large (i.e., 600), the gap narrows to 62 percent.

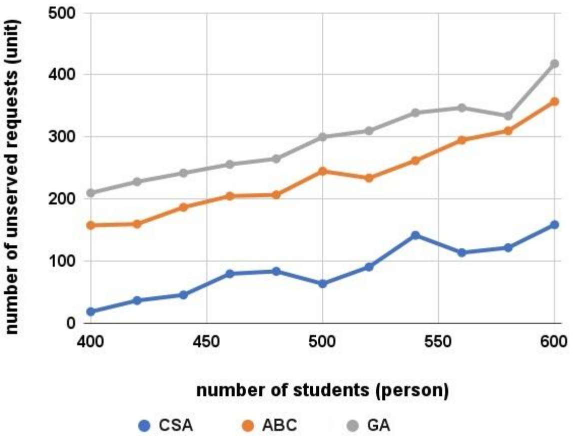

Fig. 3. Relation between the Number of Students and the Number of unserved Requests.

The second simulation is conducted to observe the relation between the average number of timeslots and the number of unserved requests. The average number of timeslots ranges from 3 to 5 timeslots. The number of students is 500 students. The result is shown in Fig. 4.

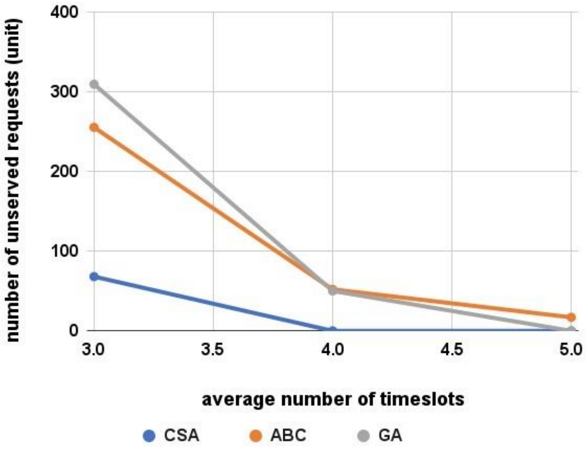

Fig. 4. Relation between the Average Number of Timeslots and the Number of unserved Requests.

In Fig. 4, it is shown that the number of unserved requests decreases due to the increase in the average number of timeslots. This condition occurs in all metaheuristic methods. The rationale is as follows. The increasing of the lecturers' timeslots means the increasing of the supply (number of classes). On the other hand, the increasing of the students' timeslots means the matching possibility increases too. When comparing among methods, the cloud theory-based simulated annealing performs as the best model. In the beginning, simulated annealing created the lowest number of unserved customers. Starting with four timeslots, the simulated annealing performs zero unserved requests. When the average number of timeslots is 3 timeslots, the artificial bee colony performs better than the genetic algorithm. The artificial bee colony has 17% fewer unsatisfied requests than the genetic algorithm. Meanwhile, when the average number of timeslots is 4 timeslots, their number of unserved requests is almost equal. Starting from 6 average timeslots, the genetic algorithm performs zero unserved requests. On the other hand, the artificial bee colony still creates a positive number of unserved requests, although its value is low (17 unserved requests).

V. Discussion

In general, these three metaheuristic models can be used to optimize the proposed course timetabling model. In the first simulation, the number of unsatisfied requests remains less than 50%. When the supply is fixed, the number of unserved requests is proportional to the demand (the number of students), as is shown in Fig. 3. On the other hand, when the demand is fixed, the increase in the supply (lecturers' number of timeslots) makes the number of unserved requests decrease. After the zero unserved requests are achieved, the increase in supply does not change the condition.

Compared among the metaheuristic techniques, the cloud theory-based simulated annealing outperforms the two other methods, the genetic algorithm, and the artificial bee colony. This performance comes from two aspects. First, in its basic form, simulated annealing is designed to achieve global optimization by avoiding local optimal traps [11]. This process is conducted by tolerating current worse solutions with a certain degree of probability during the iteration process, especially at the beginning of temperature declination [11]. Second, cloud theory based simulated annealing improves the basic simulated annealing by conducting multiple individuals (solutions) that act independently [12]. The final best solution can be selected among the population after the iteration process ends [12].

The artificial bee colony performs the second-best method. Like simulated annealing, the artificial bee colony can also avoid the local optimal trap. In an artificial bee colony, the local optimal trap avoidance is conducted during the scout-bee phase by finding new alternative solutions, i.e., diversifying the search process [24]. This phase is taken after the onlooker-bee phase and the employed-bee phase, whose objective is to intensify the solution around the current solution [24]. Unfortunately, the process of finding an absolute new solution is not conducted in every iteration. As previously stated, this concept differs from simulated annealing in that the local optimal trap avoidance can be performed with a high degree of probability in every iteration [11].

The genetic algorithm performs as the worst solution in creating a low number of unserved requests. This performance occurs because, in its basic form, the genetic algorithm cannot avoid the local optimal trap, despite the fact that it has been widely used to optimize the course timetabling problem in [4-6]. In genetic algorithms, new offspring are generated based on the best individuals as the improvement mechanism [11].

VI. Conclusion

This work has demonstrated that the proposed course timetabling model can be used in the national joint courses program that is attended by students and lecturers from different universities. This model also meets the requirements that are stated as the hard constraints. Due to the simulation process, the cloud theory-based simulated annealing performs best in minimizing the number of unserved requests. This method outperforms the two other metaheuristic methods, the genetic algorithm, and the artificial bee colony algorithm. Due to the simulation results, when the number of students is low, the number of unserved requests for the cloud theory-based simulated annealing is 91 percent lower than the genetic algorithm. When the number of students is large, this figure falls to 62 percent. This performance is achieved because of the characteristics of cloud theory-based simulated annealing in achieving global optimization by avoiding local optimal traps.

This model is developed based on several limitations. First, there is not any prioritization in the lecturers and courses selection. In certain conditions, a student prefers certain lecturers rather than other lecturers. It is because in the same course, some lecturers are more favorite or popular rather than other lecturers. For example, lecturers from higher-ranked universities may be more popular than lecturers from lower-ranked universities. A student may also prefer certain courses to other courses. It means a student may tolerate losing less preferred courses or lecturers. Based on this circumstance, the proposed model of this current work can be extended or improved by concerning this preference factor.

ACKNOWLEDGMENT

This work was funded and supported by Telkom University, Indonesia.

REFERENCES

[1] S. Dhawan, “Online learning: a panacea in the time of COVID-19 crisis”, Journal of Educational Technology Systems, vol. 49, no. 1, pp. 5-22, 2020.

[2] A. Muklason, R. G. Irianti, and A. Marom, “Automated course timetabling using tabu-variable neighbourhood search based hyper-heuristic algorithm”, Procedia Computer Science, vol. 161, pp. 656-664, 2019.

[3] I. Balan, “A new genetic approach for course timetabling problem”, Journal of Applied Computer Science & Mathematics, vol. 15, no. 1, pp. 9-14, 2021.

[4] A. Al-Majmar and T. H. Al-Shfaq, “Solving of lectures timetabling problem and automatic timetable generation using genetic algorithm”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 9, pp. 505-512, 2016.

[5] M. Assi, B. Halawi, and R. A. Haraty, “Genetic algorithm analysis using the graph coloring method for solving the university timetable problem”, Procedia Computer Science, vol. 126, pp. 899-906, 2018.

[6] D. M. Premasiril, “University timetable scheduling using genetic algorithm approach case study: Rajarata University of Sri Lanka”, Journal of Engineering Research and Application, vol. 8, no. 12, pp. 30-35, 2018.

[7] K. Zhu, L. D. Li, and M. Li, “A survey of computational intelligence in educational timetabling”, International Journal of Machine Learning and Computing, vol. 11, no. 1, pp. 40-47, 2021.

[8] C. Akkan and A. Gulcu, “A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem”, Computers and Operations Research, vol. 90, pp. 23-32, 2018.

[9] R. A. O. Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, “Practices in timetabling in higher education institutions: a systematic review”, Annals of Operations Research, vol. 275, no. 1, pp. 145-160, 2019.

[10] S. Susan and A. Bhutani, “Data mining with association rules for scheduling open elective courses using optimization algorithms”, Proceeding of International Conference on Intelligent Systems Design and Application, Cham, 2018.

[11] A. P. Engelbrecht, Computational Intelligence: An Introduction, 2nd ed., Wiley: West Sussex, 2007.

[12] C. L. Hsu, W. C. Lin, L. Duan, J. R. Liao, C. C. Wu, and J. H. Chen, “A robust two-machine flow-shop scheduling model with scenario-dependent processing times”, Discrete Dynamics in Nature and Society, article ID: 3530701, pp. 1-16, 2020.

[13] N. L. A. Aziz and N. A. H. Aizam, “A brief review on the features of university course timetabling problem”, AIP Conference Proceedings, vol. 2016, no. 1, pp. 1-7, 2016.

[14] S. L. M. Sainte, R. Jan, A. Al-Matouq, and S. Alabduhadi, “The impact of timetable on student’s absences and performance”, PloS ONE, vol. 16, no. 6, pp. 1-22, 2021.

[15] H. Alghamdi, H. Alhakami, T. Alsubait, and A. Baz, “A review of optimization algorithms for university timetable scheduling”, Engineering, Technology, & Applied Science Research, vol. 10, no. 6, pp. 6410-6417, 2020.

[16] R. Ansari and N. Saubari, “Application of genetic algorithm concept on course scheduling”, IOP Conference Series: Material Science Engineering, vol. 821, pp. 1-6, 2020.

[17] M. Lindahl, M. Sorensen, and T. R. Stidsen, “A fix and optimize matheuristic for university timetabling”, Journal of Heuristics, vol. 24, pp. 645-665, 2018.

[18] J. Xu, “Improved genetic algorithm to solve the scheduling problem of college English course”, Complexity, article ID: 7252719, pp. 1-11, 2021.

[19] N. K. Oladejo, A. Abolarinwa, S. A. Salawu, M. O. Bamiro, and A. F. Lukman, “Application of optimization principles in classroom allocation using linear programming”, International Journal of Mechanical Engineering and Technology, vol. 10, no. 1, pp. 874-885, 2019.

[20] P. Kenekayoro, “Incorporating machine learning to evaluate solutions to the university course timetabling problem”, Covenant Journal of Informatics & Communication Technology, vol. 7, no. 2, pp. 18-35, 2019.

[21] N. M. Arratia-Martinez, C. Maya-Padron, and P. A. Avila-Torres, “University course timetabling problem with professor assignment”, Mathematical Problems in Engineering, article ID: 6617177, pp. 1-9, 2021.

[22] M. M. Tavakoli, H. Shirouyehzah, F. H. Lotfi, and S. E. Najafi, “Proposing a novel heuristic algorithm for university course timetabling problem with the quality of courses rendered approach: a case study”, Alexandria Engineering Journal, vol. 59, pp. 3355-3367, 2020.

[23] H. Xuan, H. Zhang, and B. Li, “An improved discrete artificial bee colony algorithm for flexible flowshop scheduling with step deteriorating jobs and sequence-dependent setup times”, Mathematical Problems in Engineering, article ID: 8520503, pp. 1-13, 2019.

[24] B. Peng, L. Wu, Y. Wang, and Q. Wu, “Solving maximum quasi-clique problem by a hybrid artificial bee colony approach”, Information Sciences, vol. 578, pp. 214-235, 2021.