VOL. 7. NO. 2 FEBRUARY 2022
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v7i2.2262

OPTIMIZING COURSE SCHEDULING FACULTY OF ENGINEERING UNSOED USING GENETIC ALGORITHMS

Arief Kelik Nugroho[1*)], Ipung Permadi[2], Ana Romadhona Yasifa[3]

Department of Informatics, Engineering Faculty
Universitas Jenderal Soedirman
https://unsoed.ac.id/
arief.nugroho@unsoed.ac.id[1*)],ipung.permadi@unsoed.ac.id[2],ana.romadhona@mhs.unsoed.ac.id[3]

(*) Corresponding Author

Abstract— Creating an activity schedule is quite easy if the entities involved are not too many and if the entities are not bound to each other, but for larger cases, manually creating an activity schedule will take a considerable amount of time and may result in errors in the schedule or ineffectiveness in the resulting schedule. This is commonly experienced in the creation of course schedules in universities because the course data is very much and lecturers can teach several courses at once at the same time, therefore in the creation of course schedules can be done by applying genetic algorithms so that the time required in the creation of course schedules become shorter and the results obtained can be more optimal compared to the results of manually making course schedules. In this study, the optimal course schedule was obtained in the 31st generation using room data, courses, study time, lecturers, and majors so that one chromosome has 154 genes, then the length of the population is made as much as 9 individuals or chromosomes, the mutation rate is set at a value of 0.1, and the method performed in the individual selection stage is done by tournament selection method where the tournament size is set at 3. The value of fitness taken so that a schedule is said to be optimal is if the fitness value is equal to 1 because it thus indicates that there are no mistakes or problems (such as time, lecturers, conflicting rooms) that occur in the schedule.

Keywords*: Genetic Algorithm, Optimization, Scheduling.*

Abstrak—*Pembuatan jadwal kegiatan cukup mudah dilakukan apabila entitas yang terlibat tidak terlalu banyak dan apabila entitas tidak saling terikat satu sama lain, tetapi untuk kasus yang lebih besar, pembuatan jadwal kegiatan secara manual akan memakan waktu yang cukup banyak serta dapat mengakibatkan kesalahan pada jadwal atau terjadi kekurang efektifan dalam jadwal yang dihasilkan. Hal ini biasa dialami dalam pembuatan jadwal mata kuliah pada perguruan tinggi karena data mata kuliah sangat banyak serta dosen dapat mengajar beberapa mata kuliah sekaligus dan pada waktu yang tidak bersamaan, oleh karena itu dalam pembuatan jadwal mata kuliah dapat dilakukan dengan menerapkan algoritma genetika sehingga waktu yang diperlukan dalam pembuatan jadwal mata kuliah menjadi lebih singkat serta hasil yang diperoleh dapat lebih optimal dibandingkan dengan hasil pembuatan jadwal mata kuliah secara manual. Dalam penelitian ini, jadwal mata kuliah yang optimal didapat pada generasi ke-31 dengan menggunakan data ruangan, mata kuliah, waktu belajar, dosen, dan jurusan sehingga satu kromosom memiliki 154 gen, lalu panjang populasi dibuat sebanyak 9 individu atau kromosom, mutation rate diatur pada nilai 0.1, serta metode yang dilakukan dalam tahap seleksi individu dilakukan dengan metode tournament selection dimana ukuran turnamennya diatur sebesar 3. Nilai fitness yang diambil sehingga sebuah jadwal dikatakan optimal yaitu apabila nilai fitness sama dengan 1, karena dengan begitu menunjukkan bahwa tidak adanya kesalahan atau masalah (seperti waktu, dosen, ruangan yang bentrok) yang terjadi dalam jadwal tersebut.*

Kata Kunci*: Algoritma Genetika, Optimasi, Penjadwalan.*

INTRODUCTION

With the rapid development of science and technology, computers have become the backbone of various work applications with their powerful processing capabilities and fast calculation speed. They have also become the dominant force in the smart education and teacher work, especially in recent years[1]. Using computers to create sound plans and preparation for different courses can quickly adapt to different constraints and get real results[2]. (The purpose of the planning task is to

Accredited Rank 3 (Sinta 3) based on degree of the Dirjen Penguatan RisBang Kemenristekdikti
No. 28/E/KPT/2019, September 26, 2019. Published by LPPM Universitas Nusa Mandiri

first add up all the courses offered and finally to rationalize each teaching task, i.e. the course according to the teaching structure of this semester, as well as school and teacher resources, so that they optimize school resources, teachers learn rationally and students learn effectively[1]. Colleges and universities should arrange the time and location of courses and lecturers according to the structure of the teaching plan and study plan each semester. There are many rules for the course arrangement, and factors such as the teaching time, location, and lecturer must be considered. The reasonable configuration of these factors forms a professional value optimization problem with multiple constraints. At present, most colleges and universities have arranged courses manually by faculty members. Due to a large number of courses and lecturers, the required teaching locations, and multiple restrictions, many difficulties have been encountered. In response to the general needs of colleges and universities, this work takes each factor in the course scheduling problem as the input of the genetic algorithm, which is a multi-objective constrained optimization problem[3]. Many constraints and combination factors are used in the curriculum, which increases its complexity. A genetic algorithm is a parallel random search optimization algorithm that simulates natural and biological genetics and biological evolution. The algorithm established a biological evolution model and carried out corresponding calculations. The genetic algorithm can perform global optimization and parallel processing to optimize the configuration of various resources[4].

Intelligence in technology is generally called artificial intelligence, which can solve a variety of complex problems that may not be solved correctly, especially when it comes to formulating curriculum schedules, some universities still use manual methods to organize courses.

Scheduling is a combinational issue that has the limitations of conditions that must be met, therefore it becomes a complicated work that must be completed quickly, precisely, and [5] The availability of rooms, the capacity of the room, the availability of lecturers, the number of courses, the selection of time, and the number of students are absolute limitations that must be met in arranging the schedule of courses. With these restrictions, the preparation of course schedules will require thoroughness for the resulting schedule to fit, but it will certainly take a long time in arranging it and sometimes several schedules clash with each other.

In artificial intelligence, scheduling problems can be solved using genetic algorithms. A genetic algorithm is an algorithm based on the concept of evolution and gene changes in living things. The genetic algorithm was created by John Holland of the University of Michigan in 1975. Genetic algorithms are stochastic techniques or random occurrences and are based on evolutionary ideas of natural selection and genetics. Genetic algorithms are very precisely used for a variety of complex problems and are difficult to solve by conventional methods [6]. This method is categorized as a heuristic global solution finder.

The process of genetic algorithms begins with the formation of an initial population consisting of a collection of chromosomes composed of genes and is a representation of prospective completions that will be examined the actual value or fitness level, where the greater the value of fitness, the greater the likelihood of being maintained into the next population because genetic algorithms work like the principle of genetics and natural selection darwin theory that depends on the value of fitness owned by each individual. The fitness value is a value that indicates the value of chromosome toughness in adapting to problems [7]. After the formation of the initial population, the next process is an evolutionary process consisting of individual selection, crossover, and mutation. The selection process of individuals is aimed at selecting individuals as parents who will later interbreed (crossbreed) to obtain better offspring. There are several ways to make your choice, including roulette slots and tournaments. The process of crossing or crossing is a mechanism that genetic algorithms have when combining two chromosomes to produce a daughter chromosome that inherits the basic characteristics of a parent cross, acting to awaken an offspring or new daughter chromosome by replacing some information from the parent or parent's chromosome[8]. A mutation is a process of altering the sequence or composition of genes in an individual or the chromosome itself to replace genes that have been lost through the selection and crossing process [9].

Here are some scientific articles that discuss genetic algorithms [10] that discuss the application of the ma genetic algorithm in the creation of lesson schedules in a junior high school. The length of the chromosome used in this study is 2N where N is the number of hours and the room that exists and each room will be allocated subjects and teachers so that in one chromosome there are 414 genes. The perfect subject schedule will have a fitness score of 1 because the schedule has a total penalty value of the rule violated which is 0. The rule used is that if the teacher's teaching hours clash it will be given a value of 1 in the penalty rate. The selection method used in this method is a roulette wheel. Another study in 2019 by Wicaksono [11] algorithms in the placement system of SMKN students in Malang. The parameter values used are the suitability of the

CC BY NC

Accredited Rank 3 (Sinta 3) based on degree of the Dirjen Penguatan RisBang Kemenristekdikti No. 28/E/KPT/2019, September 26, 2019. Published by LPPM Universitas Nusa Mandiri

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 7. NO. 2 FEBRUARY 2022
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v7i2.2262

majors with internship vacancies, the quota of internship vacancies, student competency test scores, and the salaries of parents. Some rules must be met for each parameter and if it does not meet there are various penalty values such as unsuitable courses will get a penalty of 7, an unsuitable internship worth 1, and others. The best fitness results with a value of 0.0014286 were obtained on the number of individuals 200, the number of generations 200, the crossover percentage of 50%, and mutation 10%. Research in 2019 by Oktarina and Hajjah [8] [12]algorithms in the scheduling system of proposal seminars and thesis sessions at STIKOM Pelita Indonesia, in determining the schedule there are several rules such as in one seminar session and a session consisting of 3 lecturers, the test schedule of examiner lecturers and supervisors should not be the same and others. The number of chromosomes used in this study is 24 with each chromosome consisting of 4 criteria and each criterion has 4 sub-criteria encoded as genes, so each chromosome has 16 genes. from the processing obtained the best chromosomes in the 14th generation.

Seeing the success of genetic algorithms in handling scheduling problems in previous research, then in solving the problem of scheduling courses in this study will apply genetic algorithms and in the selection process, individuals will be used with tournament selection methods. Applying genetic algorithms to scheduling problems is expected to produce output in the form of a more effective and accurate course schedule without any problems clashing between schedules of activities and to save time in the preparation of schedules.

MATERIALS AND METHODS

Data Collection

The research data used is the data of the classroom and its capacity, day and time of learning, lecturers, courses along with several lecturers who can have and the number of students who take the course, as well as the majors along with their respective courses.

Research Stages

Figure 1. Research Stages

Explanation of research stages:

a. Literature studies

At this stage, literature collection such as books or journals or scientific articles by the research issues raised.

b. Analysis and design

At this stage, analysis of research problems is raised such as what data needs to be used and the best algorithms or methods used to deal with research problems.

c. implementation

The results of the analysis that has been done are then converted into a program using the Python programming language. Here's the flow of genetic algorithms applied in the program:

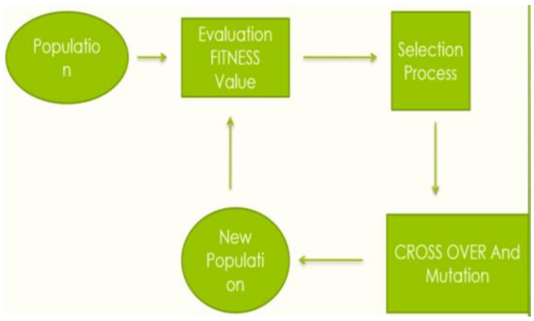

Figure 2 Genetic Algorithm Process

The process of evolution of the individual selection stage will be done using the *tournament selection* method where the tour value will be determined for the randomly selected chromosomes of a population. The best individuals in this tournament group will be selected as parents who will later be subject to the interbreeding process. The parameter used in the tournament *selection* method is the size of the tour is worth between 2 to N, where N is the sum of chromosomes in a population [13].

d. Testing and evaluation

The program results of the implementation stage are conducted testing to find out if the output results of the program are optimal and there are no problems or there are still some conflicting schedules.

RESULTS AND DISCUSSION

In the case of scheduling, the main problem that needs to be addressed is the time division of each lecturer who has more than one course and ensures that the place of learning and learning time do not collide with each other. Therefore the genetic model used consists of space as well as the day and time of learning. Courses will be arranged

P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v7i2.2262

sequentially in chromosomes followed by courses contained in those courses. The order is based on lecturers who can master the course, this is done to facilitate checking the rules related to lecturers. The length of the chromosome is the sum of all courses from each major, so the illustration of the chromosome is as follows:

Table 1 Chromosome Representation

J1						J2					
MK1			MK2			MK1			MK3		
R	D	in	R	D	in	R	D	in	R	D	in

information:

W: Learning time and day
R: Learning space
A: Courses
MK: Courses
D: Lecturer

The main thing to do in genetic algorithms is to initialize the initial population. In this process, defined the length of chromosomes used is as much as the number of all courses in each major (29 courses)multiplied by the number of existing majors (5 majors) so that the length of chromosomes is obtained as many as 154 genes and in this study a large population of 9 chromosomes in one population.

Table 2 Population Representation and Early Chromosomes

chromosome	Gen
0	TE,MKK01,E201,D2,MT13, TE,MKK02,E101,D3,MT15, TE,MKK03,C104,D4,MT14, ⋮
⋮	⋮
8	TE,MKK01,C101,D2,MT19, TE,MKK02,C103,D3,MT12, TE,MKK03,C101,D4,MT13, ⋮

After the combination of chromosomes is formed, then the calculation of errors such as courses that occupy a room that does not match its capacity, as well as errors in the time, room, and lecturers that collide schedules. Each error will be in the value of 1 and each error is accumulated in its entirety. After getting the total error in one chromosome, then done the calculation of fitness value. Here are the fitness functions used:

$$\frac{1}{(1 \times totalKesalahan)+1} \qquad (1)$$

From the above function can be concluded that the smaller the total value of the error, the smaller the fitness value, so that the optimal schedule will have a fitness value of 1 because the error that occurred does not exist or the total error is worth 0.

After calculating the fitness value of each chromosome, the next step is to check if the earliest chromosome has a fitness value equal to 1 or not, if the fitness value of the first chromosome is not equal to 1 then the program will run to the evolution stage and if the fitness value is equal to 1 then the processing has been completed and has produced an optimal schedule[14].

The first evolution stage that will be done if the fitness value is not equal to 1 is the selection stage, where the program will choose or select chromosomes to be used as parents. Parent selection is done by tournament selection method of determining the parent is done randomly from a collection of chromosomes that have been formed. The selection of chromosomes is done as much as the tournament value specified. In this study, the value of tournament used is 3, so that later obtained 3 random chromosomes that will be used as *parents.* The selected chromosomes are then sorted according to their fitness value from the smallest value to the largest value. Chromosomes with the largest fitness value are then used as parent 1 and for parent 2 is done again tournament selection process[15].

After obtaining 2 chromosomes as a parent, then the crossover process is done by swapping between selected genes on each parent chromosome. At the crossover stage, there is a crossover probability value to indicate the ratio of the resulting child in each generation to the population size. In this study, the crossover probability value used is 0.5 and because the population in this study has been initialized by 9, it is expected that 4 chromosomes out of 9 chromosomes in the population will experience crossover. Determining the point or gene to be exchanged for another parent gene is done randomly and repeatedly where if the random value is greater than 0.5 then the repeating gene to i will be exchanged for parent 1's iteration gene to i, and if the random value is less than 0.5 then the repeating gene to i will be exchanged for parent 2's iteration gene to i. This process continues until the value of i < large population.

The next stage is mutation. At this stage, the result of the crossover process will be done gene exchange with other composition genes. The mutation process is carried out as much as the number of all courses in each major and before the gene is exchanged, a random value check is performed with mutation rate. In this study mutation rate is set at 0.1, so if the random value is

Accredited Rank 3 (Sinta 3) based on degree of the Dirjen Penguatan RisBang Kemenristekdikti No. 28/E/KPT/2019, September 26, 2019. Published by LPPM Universitas Nusa Mandiri

VOL. 7. NO. 2 FEBRUARY 2022
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v7i2.2262

smaller than the mutation rate then the mutation process will not be done.

Here's a snippet of the code in the evolution section:

```
class GeneticAlgorithm:
   def evolve(self, population): return
self._mutate_population(self._crossover_population(population))
   def _crossover_population(self, pop):
      crossover_pop = Population(0)
      for i in range(NUMB_OF_ELITE_SCHEDULES):
crossover_pop.get_schedules().append(pop.get_schedules()[i])
      i = NUMB_OF_ELITE_SCHEDULES
      while i < POPULATION_SIZE:
         schedule1 =
self._select_tournament_population(pop).get_schedules()[0]
         schedule2 =
self._select_tournament_population(pop).get_schedules()[0]
crossover_pop.get_schedules().append(self._crossover_schedule(schedule1, schedule2))
         i +=1
      return crossover_pop
   def _mutate_population(self, population):
      for i in range(NUMB_OF_ELITE_SCHEDULES,
POPULATION_SIZE):
         self._mutate_schedule(population.get_schedules()[i])
      return population
   def _crossover_schedule(self, schedule1, schedule2):
      crossoverSchedule = Schedule().initialize()
      for i in range(0, len(crossoverSchedule.get_classes())):
         if (rnd.random() > 0.5):
crossoverSchedule.get_classes()[i]=schedule1.get_classes()[i]
         else:
crossoverSchedule.get_classes()[i]=schedule2.get_classes()[i]
      return crossoverSchedule
   def _mutate_schedule(self, mutateSchedule):
      schedule = Schedule().initialize()
      for i in range(0, len(mutateSchedule.get_classes())):
         if(MUTATION_RATE > rnd.random()):
mutateSchedule.get_classes()[i]=schedule.get_classes()[i]
      return mutateSchedule
   def _select_tournament_population(self, pop):
      tournament_pop = Population(0)
      i = 0
      while i < TOURNAMENT_SELECTION_SIZE:
tournament_pop.get_schedules().append(pop.get_schedules()[rnd.randrange(0, POPULATION_SIZE)])
         i += 1
      tournament_pop.get_schedules().sort(key=lambda x:
x.get_fitness(), reverse=True)
      return tournament_pop
```

Figure 3. Evaluation Process

The evolutionary process will continue if there are no chromosomes that have a fitness value equal to 1 in a population. In this study, the optimal course schedule was obtained in the 31st generation as follows:

Table 3 First Generation Evolution Results

schedule #	fitness	# of conflicts	Classes
0	1	0	TE,MKK01,C102,D25,MT14, TE,MKK02,C202,D3,MT4, TE,MKK03,C201,D4,MT11, TE,MKK05,C101,D4,MT7, TE,MKK06,C101,D6,MT1, TE,MKK07,C102,D8,MT8, TE,MKK08,E301,D7,MT6, IF,MKK05,E301,D6,MT4, IF,MKK06,C103,D6,MT8, IF,MKK07,C202,D8,MT14, IF,MKK08,C202,D7,MT20, IF ,MKK09,E101,D11,MT3, IF,MKK10,E201,D8,MT16, TS,MKK11,E301,D12,MT10, TS,MKK12,C202,D17,MT1, TS,MKK13,C201,D14,MT4, TS,MKK14,C103,D15,MT20, TS,MKK15,E301,D16,MT17, TI,MKK11,E301,D12,MT15, TI,MKK12,C202,D 13,MT16, TI,MKK16,C201,D19,MT12, TI,MKK17,C103,D20,MT1, TI,MKK18 ,E201,D21,MT10, TI,MKK19,C102,D17,MT15, TI,MKK20,C104,D18,MT14, TG,MKK21,E301,D22,MT13, TG,MKK22,C201,D23,MT16, TG,MKK23,C103,D24,MT13, TG,MKK24,C203,D25,MT8
⋮	⋮	⋮	⋮
8	0.2	4	TE,MKK01,C102,D25,MT14, TE,MKK02,C202,D3,MT4, TE,MKK03,C101,D4,MT15, TE,MKK05,C101,D4,MT7, TE,MKK06,C101,D6,MT1, TE,MKK07,C102,D8,MT8, TE,MKK08,C203,D10,MT16, IF,MKK05,C201,D4,MT2, IF,MKK06,C102,D6,MT10, IF,MKK07,C203,D9,MT13, IF,MKK08,C202,D7,MT20, IF ,MKK09,E101,D11,MT3, IF,MKK10,C201,D8,MT10, TS,MKK11,E301,D12,MT10, TS,MKK12,C202,D17,MT1, TS,MKK13,C202,D14,MT17, TS,MKK14,C102,D15,MT15, TS,MKK15,E301,D16,MT17, TI,MKK11,E301,D12,MT15, TI,MKK12,C101,D 13,MT4, TI,MKK16,E201,D19,MT18, TI,MKK17,C104,D20,MT15, TI,MKK18 ,E201,D21,MT10, TI,MKK19,C103,D17,MT7, TI,MKK20,C104,D18,MT14, TG,MKK21,E301,D22,MT13, TG,MKK22,C201,D23,MT16, TG,MKK23,C202,D24,MT20, TG,MKK24,C203,D25,MT8

Table 4 Course Schedule Program Output Results

Class #	Dept	Course (number, max # of students)	Room (Capacity)	Instructor (id)	Meeting Time (id)
0	too	Calculus (MKK01, 70)	C102 (70)	Umar (D25)	Thursday 08:50 - 10:40 (MT14)
1	too	Electrical Physics (MKK02, 20)	C202 (70)	The House (D3)	Monday 11:35am-1:55pm (MT4)
2	too	Basic Electrical Machinery (MKK03, 20)	C201 (30)	Fitria (D4)	Wednesday 09:45 - 11:35 (MT11)
3	too	Image Processing (MKK05, 40)	C101 (70)	Fitria (D4)	Tuesday 09:45 - 11:35 (MT7)

cc BY NC

Class #	Dept	Course (number, max # of students)	Room (Capacity)	Instructor (id)	Meeting Time (id)
4	too	Pattern Recognition (MKK06, 23)	C101 (70)	Bambang (D6)	Monday 07:55 - 09:45 (MT1)
5	too	Artificial Intelligence (MKK07, 40)	C102 (70)	Danang (D8)	Tuesday 11:35am-1:55pm (MT8)
6	too	Software Engineering (MKK08, 40)	E301 (70)	Christiana (D7)	Tuesday 08:50 - 10:40 (MT6)
7	IF	Image Processing (MKK05, 40)	E301 (70)	Bambang (D6)	Monday 11:35am-1:55pm (MT4)
8	IF	Pattern Recognition (MKK06, 23)	C103 (30)	Bambang (D6)	Tuesday 11:35am-1:55pm (MT8)
9	IF	Artificial Intelligence (MKK07, 40)	C202 (70)	Danang (D8)	Thursday 08:50 - 10:40 (MT14)
10	IF	Software Engineering (MKK08, 40)	C202 (70)	Christiana (D7)	Friday 15:30 - 17:30 (MT20)
11	IF	Web Design (MKK09, 10)	E101 (20)	Guntur (D11)	Monday 09:45 - 11:35 (MT3)
12	IF	Data Warehouse (MKK10, 15)	E201 (30)	Danang (D8)	Thursday 11:35am-1:55pm (MT16)
13	TS	Chemical Engineering (MKK11, 40)	E301 (70)	Hamdani (D12)	Wednesday 08:50 - 10:40 (MT10)
14	TS	Technical Economics (MKK12, 40)	C202 (70)	Muhammad (D17)	Monday 07:55 - 09:45 (MT1)
15	TS	Foundation Engineering (MKK13, 20)	C201 (30)	Janoe (D14)	Monday 11:35am-1:55pm (MT4)
16	TS	Fluid Mechanics (MKK14, 20)	C103 (30)	Kartini (D15)	Friday 15:30 - 17:30 (MT20)
17	TS	SeaPort (MKK15, 5)	E301 (70)	Lukman (D16)	Friday 07:55 - 09:45 (MT17)
18	Ti	Chemical Engineering (MKK11, 40)	E301 (70)	Hamdani (D12)	Thursday 09:45 - 11:35 (MT15)
19	Ti	Technical Economics (MKK12, 40)	C202 (70)	Priest (D13)	Thursday 11:35am-1:55pm (MT16)
20	Ti	Introduction to Industrial Engineering (MKK16, 20)	C201 (30)	Reason (D19)	Wednesday 11:35am-1:55pm (MT12)
21	Ti	Ergonomics (MKK17, 20)	C103 (30)	Praise (D20)	Monday 07:55 - 09:45 (MT1)
22	Ti	Safety Systems (MKK18, 20)	E201 (30)	Qonita (D21)	Wednesday 08:50 - 10:40 (MT10)
23	Ti	Work Productivity (MKK19, 5)	C102 (70)	Muhammad (D17)	Thursday 09:45 - 11:35 (MT15)
24	Ti	Industrial Energy Management (MKK20, 20)	C104 (30)	Nike (D18)	Thursday 08:50 - 10:40 (MT14)
25	TG	Basic Geology (MKK21, 20)	E301 (70)	Robby (D22)	Thursday 07:55 - 09:45 (MT13)
26	TG	Indonesian Geology (MKK22, 15)	C201 (30)	Salman (D23)	Thursday 11:35am-1:55pm (MT16)
27	TG	Hydrogeology (MKK23, 10)	C103 (30)	Dance (D24)	Thursday 07:55 - 09:45 (MT13)
28	TG	Mineral Deposits (MKK24, 20)	C203 (30)	Umar (D25)	Tuesday 11:35am-1:55pm (MT8)

CONCLUSION

From the research conducted, it can be concluded that the first generation in this study has been able to produce an optimal lecture schedule, therefore genetic algorithms can be applied in handling the problem of optimization of course schedules so that a more optimal lecture schedule is produced without the problem of conflicting schedules, effective and accurate without takes a long time in the process of schedule making. However, to avoid getting caught up in the iteration of processing that can take a long time if the program continues to run because it has not found an optimal schedule or the best *fitness* value, it takes limitations or conditions that can stop the work of the program when facing the problem.

REFERENCE

[1] J. Xu, "Improved Genetic Algorithm to Solve the Scheduling Problem of College English Courses," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/7252719.
[2] I. K. W. I Gusti Ayu Desi Saryanti, "Penerapan Metode Algoritma Genetika untuk Penjadwalan Mengajar," *Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 8, no. 1, 2017.
[3] S. N. Jat and S. Yang, "A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling," *Journal of Scheduling*, vol. 14, no. 6, 2011, doi: 10.1007/s10951-010-0202-0.
[4] W. Wen-Jing, "Improved adaptive genetic algorithm for course scheduling in colleges and universities," *International Journal of*

Accredited Rank 3 (Sinta 3) based on degree of the Dirjen Penguatan RisBang Kemenristekdikti No. 28/E/KPT/2019, September 26, 2019. Published by LPPM Universitas Nusa Mandiri

VOL. 7. NO. 2 FEBRUARY 2022
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v7i2.2262

Emerging Technologies in Learning, vol. 13, no. 6, 2018, doi: 10.3991/ijet.v13i06.8442.

[5] A. Nugroho, W. Priatna, and I. Romli, "Implementasi Algoritma Genetika untuk Optimasi Penjadwalan Mata Kuliah," *Jurnal Penelitian Teknik Informatika*, vol. 1, no. 2, pp. 188–194, 2018.

[6] A. Jain, S. Jain, and P. K. Chande, "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable," *International Journal of Innovation, Management and Technology*, vol. 1, no. 3, pp. 248–251, 2010.

[7] A. S. Wicaksana, B. D. Setiawan, and B. Rahayudi, "Algoritma Genetika untuk Optimasi Fuzzy Time Series dalam Memprediksi Kepadatan Lalu Lintas di Jalan Tol," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 3, pp. 1063–1071, 2018.

[8] D. A. Suprayogi and W. F. Mahmudy, "Penerapan Algoritma Genetika Travelling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry," *Jurnal Buana Informatika*, vol. 6, no. 2, pp. 121–130, 2015.

[9] L. Paranduk, A. Indriani, M. Hafid, and Suprianto, "Sistem Informasi Penjadwalan Mata Kuliah Menggunakan Algoritma Berbasis Web," *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, pp. E46–E50, 2018.

[10] N. L. G. P. Suwirmayanti, I. M. Sudarsana, and S. Darmayasa, "Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran," *Journal of Applied Intelligent System*, vol. 1, no. 3, pp. 220–233, 2016.

[11] S. A. Wicaksono, "Optimasi Sistem Penempatan Magang Menerapkan Algoritme Genetika," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 1, p. 17, 2019, doi: 10.25126/jtiik.201961950.

[12] D. Oktarina and A. Hajjah, "Perancangan Sistem Penjadwalan Seminar Proposal Dan Sidang Skripsi Dengan Metode Algoritma Genetika," *JOISIE Journal Of Information System And Informatics Engineering*, vol. 3, no. 1, pp. 32–40, 2019.

[13] D. Hermawanto, "Beberapa Metode Seleksi Algoritma Genetika | denny...," May 11, 2007. https://dennyhermawanto.wordpress.com/2007/05/11/beberapa-metode-seleksi-algoritma-genetika/ (accessed Jun. 16, 2021).

[14] A. K. Nugroho, "Image Quantization in Psoriasis Using K-Mean Clustering," *Conference SENATIK STT Adisutjipto Yogyakarta*, vol. 4, 2018, doi: 10.28989/senatik.v4i0.162.

[15] I. Permadi and A. K. Nugroho, "Klasifikasi Citra Menggunakan Kombinasi Jaringan Syaraf Tiruan Model Perceptron dan Algoritma One vs Rest," *INOVTEK Polbeng - Seri Informatika*, vol. 4, no. 2, 2019, doi: 10.35314/isi.v4i2.1062.

cc BY NC

Accredited Rank 3 (Sinta 3) based on degree of the Dirjen Penguatan RisBang Kemenristekdikti No. 28/E/KPT/2019, September 26, 2019. Published by LPPM Universitas Nusa Mandiri