Genetic Algorithm for Scheduling Courses

Gregorius Satia Budhi, Kartika Gunadi, Denny Alexander Wibowo

Petra Christian University, Informatics Department
Siwalankerto 121-131, Surabaya, East Java, Indonesia
e-mail: greg@petra.ac.id, kgunadi@petra.ac.id

Abstract. In the university, college students must be register for their classes, and there still many college student that was confused on how to make a good classes schedule for themselves. Mainly because of many variables and considerations to be made, for examples, they have to consider how hard the classes they are going to take, and also, they still have to consider their exam schedules and also their availability time as well. Genetic Algorithm is one of many methods that can be used to create a schedule. This method determines the best schedule using fitness cost calculation which can compare the quality of one schedule against the other. Then, using crossover, mutation, and elitism selections, we can determine better schedules. Based on the result of the survey held before, 70% of the respondents gave point 4 and 30% of the respondents gave point 5 out of 5 for the quality of the schedule made using this applications.

Keywords: Artificial Intelligence, Genetic Algorithm, Courses Schedule

1 Introduction

Schedule is very important to manage an activity, especially when the activity was carried out in a large organization and held for long-term time or routine. Within the scope of the university, the course schedule also is critical There are still students who are confused when creating their own class schedules, because a lot of consideration, for example: what subjects can be taken without clashing with another schedule, a student who wants to the day - certain the day No lectures, due to personal reasons or other student affairs, student who wants to take courses that would like to correspond with their fields of specialization, and so forth. These things make them have trouble making a personal course schedule. Through these problems, we conduct research on a system that can help perform automatic scheduling lectures for students by the day and the course desired by the student. The method used for the manufacture of automatic schedule is Genetic Algorithm (GA). The reason is because GA is a method that can find a solution that is both complex problems.

adfa, p. 1, 2011.

2 Genetic Algorithm

Genetic Algorithm is a method to search an optimal solution for a problem. The method will find a good solution by crossovers a possible solution with another solution to create new solutions. After that method will mutate the new solutions so that they are have parts of solution from the parents but not really same with the parent. The process begin with the creation of random population of valid solutions / chromosomes, then GA will count the fitness costs of each chromosomes in the population. After that two chromosomes will be chose to crossover and produce offspring. Then the offspring will be mutated. This process will be repeated till the stop condition was reached [1].

Given a clearly defined problem to be solved and a binary string representation for candidate solutions, a basic GA can be represented as following major steps [2]:

1. Represent the problem variable domain as a chromosome of a fixed length, choose the size of a chromosome population N, the crossover probability p_c and the mutation probability p_m.
2. Step 2: Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.
3. Randomly generate an initial population of chromosomes of size N:

$$x_1; x_2; ...; x_N \qquad (1)$$

4. Calculate the fitness of each individual chromosome:

$$f(x_1) ; f(x_2); ...; f(x_N) \qquad (2)$$

5. Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. Highly fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.
6. Create a pair of offspring chromosomes by applying the genetic operators – crossover and mutation.
7. Place the created offspring chromosomes in the new population.
8. Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population, N.
9. Replace the initial (parent) chromosome population with the new (offspring) population.
10. Go to Step 4, and repeat the process until the termination criterion is satisfied.

2.1 Selection Criteria

The commonly used chromosome selection technique is the roulette wheel selection [3, 4]. It is like spinning a roulette wheel where each chromosome has a segment on the wheel proportional to its fitness. The roulette wheel is spun, and when the arrow comes to rest on one of the segments, the corresponding chromosome is selected. Each chromosome is given a slice of a circular roulette wheel. The most fits chromosomes occupy the largest areas, whereas the least fits have much smaller segments in the roulette wheel. The area of the slice within the wheel is equal to the chromosome fitness ratio. To select a chromosome for mating, a random number is generated in the interval (0, 100), and the chromosome whose segment spans the random number is selected [2]. For the example you could see in Table 1 and Figure 1.

Table 1. Example of Chromosomes and the fitness (Source: Negnevitsky, 2005).

Chromosome label	Chromosome fitness	Fitness ratio, %
X1	36	16.5
X2	44	20.2
X3	14	6.4
X4	14	6.4
X5	56	25.7
X6	54	24.8

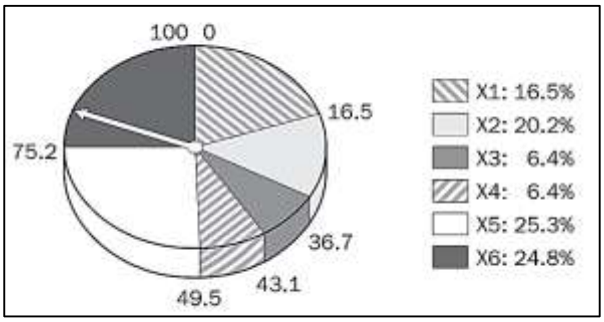


Fig. 1. Example of the slices in roulette wheel selection (Source: Negnevitsky, 2005)

2.2 Reproduction

There are several methods used in the GA Reproduction one of them is crossover. Crossover is the crosses of at least 2 solutions, where through parts crossed, will generate new solutions. Through this crosses GA will add solutions and variations that are useful later in the scoring stage when searching for a better solution [5]. The example of the crossover could be seen in Figure 2.

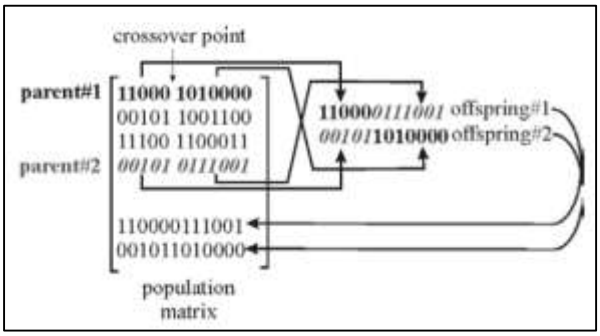


Fig. 2. Crossover in binary chromosome (Source: Haupt, R.L., Haupt, S.E., 2004)

Besides the crossover, there is another way to complete reproduction, namely mutation. Mutation is a method that converts a portion of the solution. Although altered a little bit, it can create different variations resulting in different solutions and even new solution. One method used in this mutation is shuffle. [5]. Mutation role is to provide a guarantee that the search algorithm is not trapped on a local optimum [6]. The example of the mutation could be seen in Figure 3.

$$\boxed{00101100000001 \Rightarrow 00101\boldsymbol{0}00000001}$$

Fig. 3. Mutation in a binary chromosome (Source: Haupt, R.L., Haupt, S.E., 2004)

3 Courses Enrolling on Study Plan Registration time

Students will enroll subjects which wants to be taken at the beginning of the semester, during the study plan registration (SPR/PRS). PRS periods are divided into three times, namely the PRS-1, PRS-2 and PRS-3. During the PRS-1, after receiving Study Results Card (SRC/KHS), and get permission to register from faculty trustee, students can enroll subjects that have been opened by the department. For example, the list of courses that are opened by PCU Informatics department could be seen in Figure 4.

HARI	JAM	SAMPAI	NOVAK	NAMA	KLS
SENIN	07.30	10.30	TF4303	APLIKASI SISTEM PAKAR	
	08.30	11.30	TF4343	TEKNOLOGI WEB	B
	09.30	11.30	TF4205	DASAR SISTEM KOMPUTER	A
	09.30	11.30	TF4259	METODOLOGI PENELITIAN	B
	09.30	11.30	TF4273	BASIS DATA LANJUTAN	C
	13.30	16.30	TF4220	PENG. MANAJEMEN & BISNIS	
	13.30	15.30	TF4211	KALKULUS I	B
	13.30	16.30	TF4225	PAKET PROGRAM APLIKASI	C
	14.30	17.30	TF4241	SISTEM INFORMASI	A
	14.30	17.30	TF4413	E - BUSINESS	B
	15.30	18.30	TF4361	PEMROGRAMAN APLIKASI JAVA	
	15.30	18.30	TF4363	PEMROGRAMAN OPEN SOURCE	
	15.30	18.30	TF4337	SISTEM LOGIKA FUZZY	
	15.30	17.30	TF4211	KALKULUS I	C
	17.30	19.30	TF4211	KALKULUS I	D
	18.30	20.30	TF4273	BASIS DATA LANJUTAN	A
SELASA	07.30	09.30	TF4223	ALJABAR LINIER DAN MATRIKS	
	07.30	10.30	TF4351	SISTEM INFORMASI GEOGRAFIS	
	07.30	10.30	TF4343	TEKNOLOGI WEB	A
	07.30	09.30	TF4204	ALGORITMA DAN PEMROGRAMAN	E
	08.30	11.30	TF4415	GRAFIKA KOMPUTER	

Fig. 4. The list of courses that are opened for one semester

Then in Figure 5 you could see the example of courses that have been taken by students that called the transcript.

Kode MK	Nama Mata Kuliah	SMT	SKS	Nilai
TF4205	DASAR SISTEM KOMPUTER	1-08/09	2	B
DU4161	ETIKA	1-08/09	2	A
TF4216	RANGKAIAN DIGITAL	1-08/09	3	C
TF4218	PENGANTAR MANAJEMEN	1-08/09	2	B
TF4211	KALKULUS I	1-08/09	3	C
TF4204	ALGORITMA DAN PEMROGRAMAN	1-08/09	4	A
DU4167	FILSAFAT AGAMA	1-08/09	2	B+
TF4233	LOGIKA MATEMATIKA	2-08/09	3	B
DU4168	PANCASILA DAN KEWARGANEGARAAN	2-08/09	2	A
TF4235	PEMROGRAMAN BERORIENTASI OBYEK	2-08/09	3	B+

Fig. 5. Student transcript example

Based on the example of courses offered (Figure 4) and student transcript (Figure 5), the list of courses that can be taken by the student in the next semester could be seen in Table 2.

Table 2. The example of courses which could be taken by student base on Figure 4 and 5

Course Code and Name	Class	Type	Prerequisite
TF4251 – Komunikasi Data	A,B,C	Mandatory	K. RD
TF4219 – Struktur Data	A	Mandatory	K. PBO
TF4229 – Basis Data	A	Mandatory	-

For courses with prerequisites "Had attended", if the student has taken the prerequisite courses but did not passed, he could take courses with these prerequisites. But for courses with prerequisites "Should be passed", the courses prerequisites should be passed before the desired courses taken. Courses who had been taken by a student, still could be taken again even after passed, with the aim to improve the credit point.

For courses with 'Elective' type, it can be taken if it meets prerequisites and the minimum number of credits that have been completed by the student is 60 credits. For courses with 'Mandatory' type, it could be taken when the student have met the prerequisites of the courses.

When registering on the PRS, students are given choice of priority 1 and 2. For example, if a student has a value of 20 for priority 1, then the student should only enroll 20 Semester Credit Units (SCU/SKS) with priority 1, and the others are priority 2. The priority 1 is very important because the chance it is likely to be accepted in the class larger than the one with priority 2.

During the PRS-2 students could revise or add more courses to be taken up to a maximum limit of credits. In addition, students could cancel courses which has received in PRS-1. While the PRS-3 is the time to cancel courses that have been received in the PRS-1 and PRS-2.

4 Genetic Algorithm Design

4.1 Chromosome Design

Chromosome is an important factor in the genetic algorithm, in which the design of chromosomes will affect how the crossover implemented. Form of chromosome could be seen in Figure 6.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Day 0	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Day 1	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Day 2	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Day 3	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Day 4	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Day 5	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Fig. 6. Chromosome Design

The chromosome has day index starting from zero to the fifth indexes. Where in each day they have hour index starting from zero to the fourteen indexes.

Day index only have six days, because on the day of Sunday is certainly no lectures. As for the hour index in each day, containing fourteen ranging from 07.30 to 22.30 with a time range of 1 hour. For example of chromosomes that have been filled could be seen in Figure 7.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Day 0	[2]	[2]	[2]	[0]	[0]	[6]	[6]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
Day 1	[0]	[0]	[49]	[49]	[0]	[0]	[0]	[0]	[10]	[10]	[0]	[0]	[0]	[0]	[0]
Day 2	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
Day 3	[0]	[0]	[0]	[0]	[0]	[0]	[19]	[19]	[19]	[0]	[0]	[0]	[0]	[0]	[0]
Day 4	[0]	[33]	[33]	[33]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
Day 5	[0]	[0]	[0]	[0]	[44]	[44]	[44]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]

Fig. 7. Example of chromosomes that have been filled

The values present in a chromosome are the id of courses that opened in a semester. Complete data from the ids are stored in a database table with the design as shown in Table 3.

Table 3. Id Class information

ID class	49
Course name	Struktur Data
Class	A
SKSk	2
SKSp	1
SKSs	0
SKSr	0
SKS prerequisites	0
Course prerequisites	Pass PBO, Pass AP
Course Start	08.30
Course Finish	10.30
Exam Schedule	Day 3, 10.30
Specialization Course	No

4.2 Crossover and Mutation Design

In this study, crossover design was made with the rule that a slot of a particular course can only be subjected to a cross for all of the time slots of the course, and crossed to all of the time slots of another course in the other chromosome. After that the crossover results need to be checked whether valid or did not. If invalid, then the result was not used in subsequent process. The example could be seen in Figure 8.

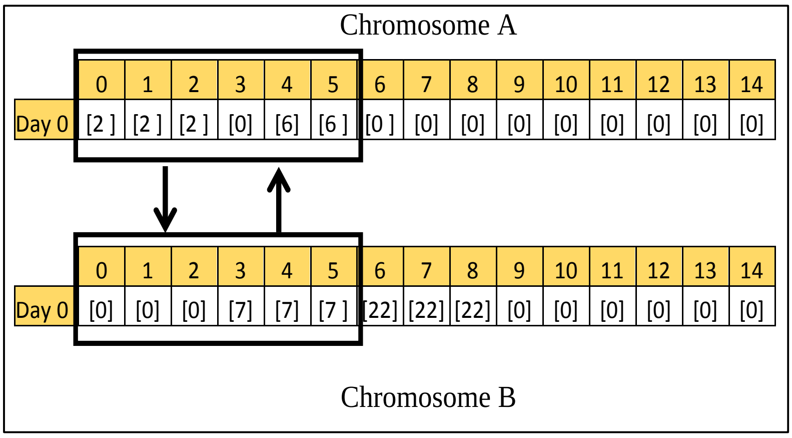


Fig. 8. The example of crossover

Whilst the mutation is done by randomly selecting two courses that have the same length of time in the one of crossover result chromosome, the offspring, and then swap their positions.

4.3 Fitness Cost

To determine the quality of the chromosome that is created, fitness cost calculation is required. Here are the fitness cost calculation formulas of a chromosome. We adopt the fitness cost calculation formulas from the similar formulas that was used by researchers in the previous work [7].

- Fitness Cost based on Credit Point Units/SKS (FCSKS)

$$FCSKS = \text{totalCreditPointUnit} * 30 \quad (3)$$

- Fitness Cost based on the semester to be taken (FCSM)

$$FCSM = \sum_{i=1}^{m} courseSemester[i] \quad (4)$$

Where:
If the course is mandatory for next semester:
    courseSemester = (courseSemester [i] - semesters) * (-5)
If the course is mandatory for the last semester:
    courseSemester = (semesters - courseSemester [i]) * (200)
If the course is mandatory for the current:
    courseSemester = 100
m = number of existing courses

- Fitness Cost based on curriculum (FCKUR)

$$FCKUR = \sum_{i=1}^{m} course\text{Curriculum}[i] \qquad (5)$$

Where:
If the elective courses taken curriculum matches the user specialization:
    courseCurriculum = 175
If the elective courses taken curriculum does not match the user specialization:
    courseCurriculum = -75
m = number of courses in the chromosome

- Fitness Cost based on the interval between courses (FCJMK):

$$FCJMK = \sum_{i=1}^{n} dayInterval[i] \qquad (6)$$

Where:
dayInterval = 300 - (interval between courses * 90)
n = number of day

- Fitness Cost based on courses to be taken (FCMKI)

$$FCMKI = \sum_{i=1}^{n} \text{ValueDesireCourse}[i] \qquad (7)$$

Where:
n = number of existing courses on chromosome
If the user wishes 'Want to Take':
    ValueDesireCourse = 100
If the user wishes 'Extremely Want to Take':
    ValueDesireCourse = 1000
If the user wishes 'Less Want to Take':
    ValueDesireCourse = -200

- Fitness Cost based on the time desired by the user (FCWU)

$$FCWU = \sum_{i=1}^{n} courseDay\text{Desired}[i] \qquad (8)$$

$$course\text{DayDesired} = \sum_{j=1}^{k} timeValue[j] \qquad (9)$$

Where:
n = number of day
k = 3, on the morning: k = 1, on the afternoon: k = 2, on the night: k = 3
If in accordance with the expected time:
    timeValue = number of hours filled * 200
If not in accordance with the expected time:
    timeValue = number of hours filled * -200

- Fitness Cost based on the maximum number of courses per day that is desired by the user (FCMMK)
  - If user desirability ≥ number of courses,

$$FCMMK = \sum_{i=1}^{n} \frac{numberOfCourses\,[i]}{userDesirability[i]} * 300 \qquad (10)$$

– If user desirability ≥ number of courses

$$FCMMK = \sum_{i=1}^{n} \text{userDesirability[i]} - numberOfCourses\text{[i]} * 500 \quad (11)$$

Where:
n = number of day

- Furthermore, the formula to calculate overall fitness cost of a chromosomme is as follows:

  Fitness Cost = FCSKS * w1 + FCSM * w2 + FCKUR * w3 + FCJMK * w4 + FCMKI * w5 + FCWU * w6 + FCMMK * w7 (12)

  Where:
  w1 to w7 = weight of each fitness cost, value: 0...1.

5 Implementation and experiment results

System workflow could be seen in Figure 9.

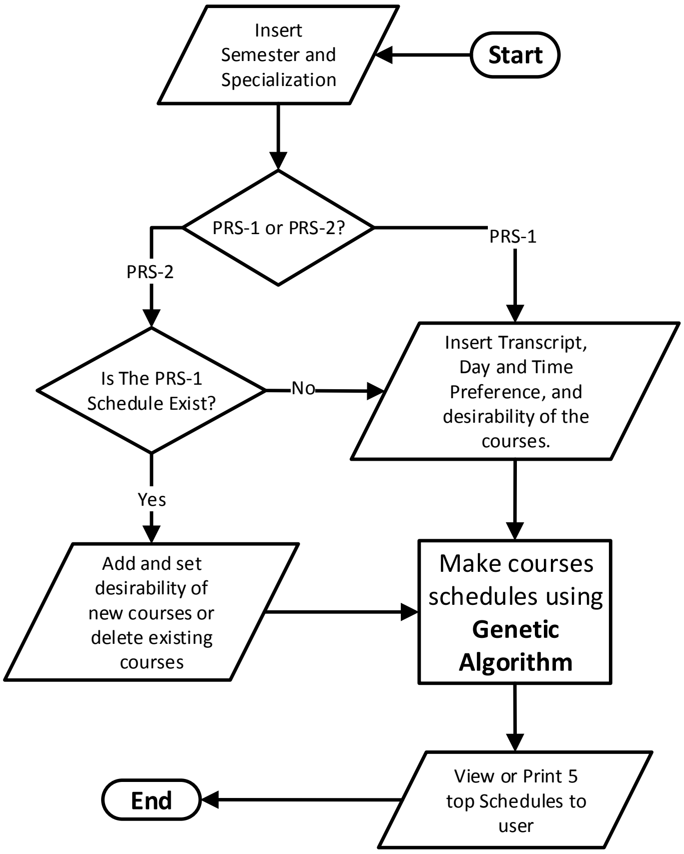


Fig. 9. System Workflow

System starts with a login form to identify the user of the application. Furthermore, the user can enter semester to be processed and areas of specialization are taken, as shown in Figure 10.

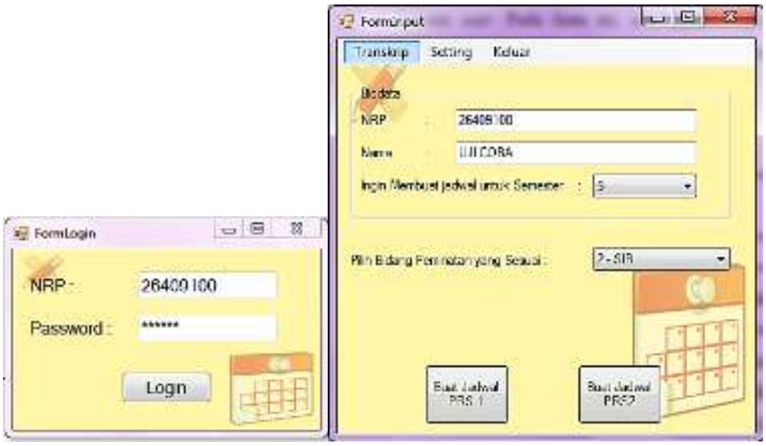


Fig. 10. Form Login and form setting semester and specialization

Later the user can choose to process the PRS-1 or PRS-2. Next will be displayed to the user some forms in sequence, as shown in Figure 11 to 12. Here the user can enter the latest transcript, day and time he desire and the desirability settings for all courses that could be taken for the next semester.

Fig. 11. Form to insert the latest transcript

Fig. 12. Form to choose preferences days and hours of courses, and the number of courses/day

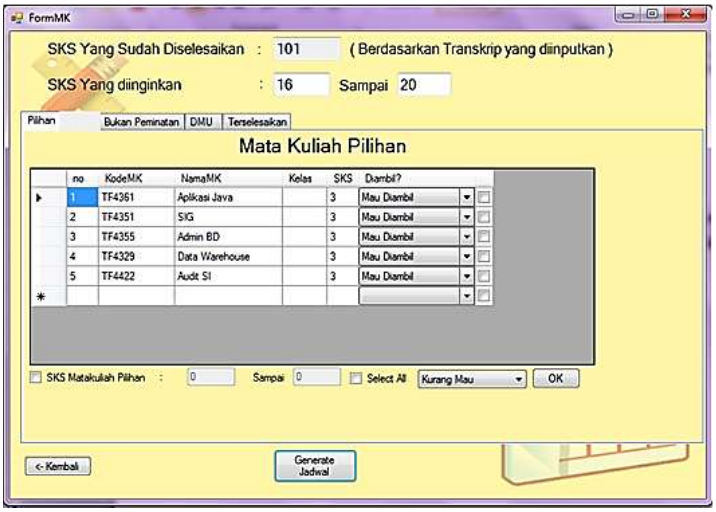


Fig. 13. Form to determine the desirability of all the courses that can be taken

Hereinafter the user can press a button to generate schedules using Genetic Algorithm. Once the process is complete then to the user would be presented 5 best schedules. These 5 best schedules are taken from 5 chromosomes with the highest fitness costs on the last population. The scheduling results form and also the look to be printed (HTML files) could be seen in Figure 14 and Figure 15. In addition, before starting the process of scheduling, user can change the settings of variables required by the GA. Form to change the settings can be seen in Figure 16.

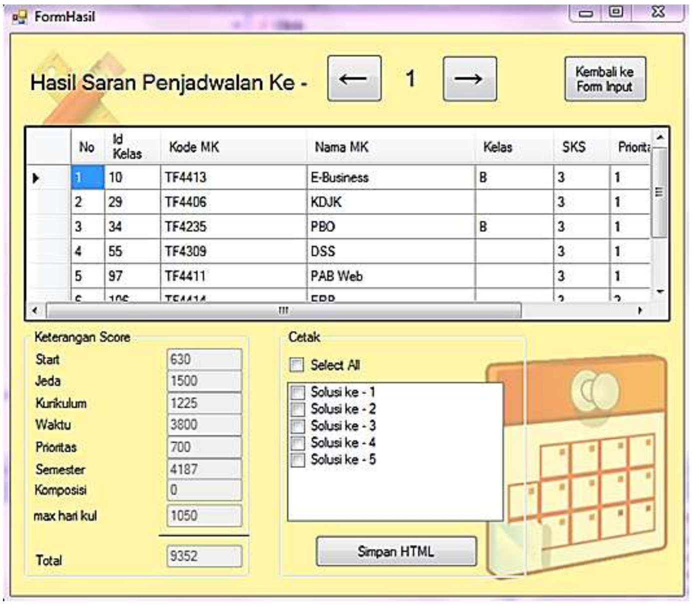
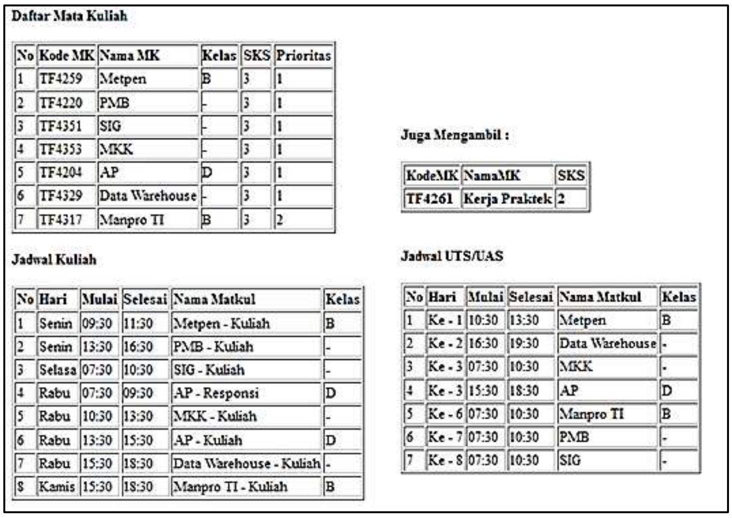


Fig. 14. Form to view 5 best the results of automatic courses scheduling



Daftar Mata Kuliah

No	Kode MK	Nama MK	Kelas	SKS	Prioritas
1	TF4259	Metpen	B	3	1
2	TF4220	PMB	-	3	1
3	TF4351	SIG	-	3	1
4	TF4353	MKK	-	3	1
5	TF4204	AP	D	3	1
6	TF4329	Data Warehouse	-	3	1
7	TF4317	Manpro TI	B	3	2

Juga Mengambil :

KodeMK	NamaMK	SKS
TF4261	Kerja Praktek	2

Jadwal Kuliah

No	Hari	Mulai	Selesai	Nama Matkul	Kelas
1	Senin	09:30	11:30	Metpen - Kuliah	B
2	Senin	13:30	16:30	PMB - Kuliah	-
3	Selasa	07:30	10:30	SIG - Kuliah	-
4	Rabu	07:30	09:30	AP - Responsi	D
5	Rabu	10:30	13:30	MKK - Kuliah	-
6	Rabu	13:30	15:30	AP - Kuliah	D
7	Rabu	15:30	18:30	Data Warehouse - Kuliah	-
8	Kamis	15:30	18:30	Manpro TI - Kuliah	B

Jadwal UTS/UAS

No	Hari	Mulai	Selesai	Nama Matkul	Kelas
1	Ke - 1	10:30	13:30	Metpen	B
2	Ke - 2	16:30	19:30	Data Warehouse	-
3	Ke - 3	07:30	10:30	MKK	-
4	Ke - 3	15:30	18:30	AP	D
5	Ke - 6	07:30	10:30	Manpro TI	B
6	Ke - 7	07:30	10:30	PMB	-
7	Ke - 8	07:30	10:30	SIG	-

Fig. 15. The display of HTML files to print the schedule

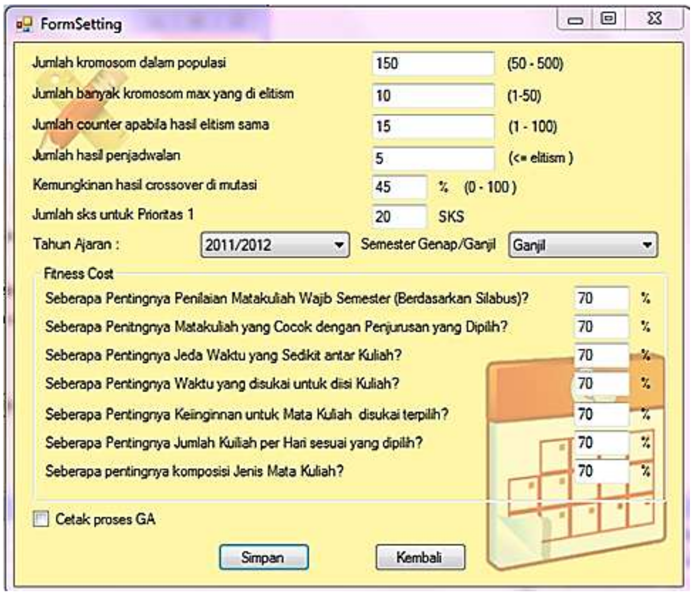


Fig. 16. Form setting for Genetic Algorithm process

To test the results of the system, especially the results of the courses schedule that is generated automatically, we use several kinds of tests. The first test is testing the effect of the maximum population against the processing time, and the best fitness cost value. For this test, setting the mutation probability is 0.3. The test results could be seen in Table 4.

Table 4. Tests on the maximum population size

Maximum Population	Running Times (millisecond)		Average of Fitness Costs
	Generate Initial Chromosomes	Genetic Algorithm Processes	
50	5962.8	3266.6	4302.4
150	15070.8	17938.4	4782.4
300	23283.2	32652.8	5297.4

From the test results it can be seen that the greater the maximum limit of the population, the longer the process takes time, yet the average value of the best fitness costs also increased. This is because more and more of the population there, will be the more diverse opportunities to generate chromosomes, so that eventually the fitness cost values of the chromosomes will increase.

The second test is the test of the effects of increased mutation probability values allowed on offspring chromosomes. For this test, the maximum population size be set at 150 chromosome. The test results can be seen in Table 5.

Table 5. Tests on the mutation probability values

Mutation Probability	Running Times (millisecond)		Average of Fitness Costs
	Generate Initial Chromosomes	Genetic Algorithm Processes	
0.1	11614	11935	4532.8
0.4	14621.4	12650	4712
0.7	15942.2	19960.6	5176

From the test results it can be seen that the greater the probability of mutation, the longer the time required for the process, but the average fitness value of a chromosome best cost also increased. This is because the higher the chance of mutation, the more diverse the resulting chromosome will be are greater, so that eventually the fitness cost value of the best chromosome will increase.

In addition we also did some other tests, such as testing the maximum number of chromosomes are copied by elitism method from the old population to the new population. We also tested the maximum number of 'elitism counter', this counter to count how many the sequences of populations will produce the same best fitness cost values, before the program stop the process. Of all the tests performed at last we obtained the best values of each setting. These values are used as the default values setting of the system. The amount of the default settings and the testing results using default settings can be seen in Table 6 and Table 7.

Table 6. Default *setting*

Setting Type	Value
Maximum population	150
Max Elitism	15
Elitism Counter	15
Mutation Probability	0.7

Table 7. Testing results using default setting

Number of Tests	Running Times (millisecond)		Scores
	Generate Initial Chromosomes	Genetic Algorithm Processes	
1	23523	34653	5317
2	17116	18932	5317
3	17177	34653	5317
4	18592	32155	5269
5	33025	34090	5317

Average:	21886.6	30896.6	5307.4

Of testing by using the default settings could be seen that the GA produces the best fitness cost value greater compared to other settings that have been tested. Moreover, it can be seen that in general the system will be convergence on a particular chromosome.

The final testing is testing in the form of questionnaires to potential users, in example students majoring in Informatics department of Petra Christian University. The results summary of these questionnaires are as follows: As many as 60% of respondents gave scores of 4 and 40% gave scores of 5, of the scale from 1 (worst) to 5 (best), for ease of use of this application. From these results it can be concluded that the user interface of the application is easy to use. Meanwhile, for the question about the quality of the generated schedules, 70% of respondents gave scores of 4 and 30% gave scores of 5, of scale from 1 (worst) to 5 (best). Therefore, it can be concluded that in general the respondents felt be helped by the application that could generate courses schedule automatically.

6 Conclusion

From the tests it can be concluded that the automatic scheduling system is well made. Processing speed is also good, averaging less than 1 minute. From the test results on the potential users can be concluded that the interface has a good design and user friendly. In addition, the automatically generated class schedules are also correct and in accordance with the expectations of users.

References

1. Mitchell, M.: An Introduction to Genetic Algorithms. United States of America: Massachusetts Institute of Technology. (1996).
2. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems 2nd Ed. Pearson Education Limited. (2005)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA. (1989).
4. Davis, L.: Handbook on Genetic Algorithms. Van Nostrand Reinhold, New York. (1991).
5. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms 2nd Ed. United States of America: John Wiley & Sons, Inc. (2004).
6. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor. (1975).
7. Budhi, G.S., Handojo, A., Soloment, B.: The Use of compact-Genetic Algorithm (cGA) for Petra Christian University Classroom Scheduling Optimization. In: Proc. National Seminar on Information Technology. (2009).