15

Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable

Samuel Lukas, Arnold Aribowo and Milyandreana Muchri
Faculty of Computer Science, Universitas Pelita Harapan, Indonesia

1. Introduction

Almost all education institutes have problem concerning with scheduling, especially university. Many things have to be considered in order to arrange schedule. One of them is availability of lecturers. Not all lecturers are available at any time. Some of them are just available in some time. Therefore, when schedule is arranged, this thing has to be considered. The other things are number of classes and courses offered. Number of classes and courses in university timetable are many. Room availability is other thing, budgeting and many others.

In Indonesian education system, undergraduate students can earn their degree after finishing at least 144 semester credit units. For one unit course, student should attend 50 minutes in class, added by 50 minutes for homework and another 50 minutes for independent activity. In average one course consists of 3 semester credit units. Therefore, to finish their study, students should take about minimum 45 courses. In one semester, students take maximum 24 units and minimum 12 units. Normally, it takes about 4 years of study for a bachelor degree. It means that each semester students have to take minimum 6 courses and maximum about 8 courses unless for the last semester student only take maximum 14 units, one of them is final project which is counted maximum 6 units. Excellence students will finish their study for about 7 semesters. It means that each semester, in average they have to take about 22 units.

It is obvious that within the same semester, all courses have to be scheduled differently one and another so that student can take the course without any overlapping schedule. All of these courses are registered as a group. Since there are four years of study, then the number of different course groups is minimum four for one department.

For a certain department, the number of students in one batch is very big and it is impossible to schedule them for a certain course in one class. Therefore, parallel class most likely will happen. Suppose there are 100 students will take a certain course in the same semester. Since there are only maximum 25 students in one class, then for that course will be opened 4 parallel classes. The schedule of that parallel class does not have to be the same. It depends on the lecturer availability time. In addition, there are also possibilities for a certain

course that some classes are merged into one class. Furthermore, there are not one to one mapping between lecturer and courses. One lecturer can teach a number of courses. It will cause making the time table harder.

Universitas Pelita Harapan timetable consists of about 38 departments. The number of students intake each year is about 2000 students. The constraints of the time table are firstly, there are 10 hours lecture time a day and five days a week. Secondly, there are two types of lecture, fulltime and part time lecturer. The part time lecturer maximum is scheduled only 6 units a week, whereas fulltime is maximum 12 units. There is no constraint with the room. However, for some certain courses, there are also laboratory works to be scheduled differently. It is also making the time table harder.

Genetic Algorithm (GA) was powerful to solve assignment problem (Lukas et al, 2005). GA was also used for creating university exam timetable (Burke, et all, 1994). Heuristic search was used for solving scheduling (Joshua and Graham, 2008). This chapter proposes a method for solving this time table problem by using genetic algorithm combined with heuristic search. The role of genetic algorithm is to determine the sequence of all courses to be scheduled in one group, whereas the role of heuristic search is to determine time slots used to schedule the courses (Thanh, 2007).

This chapter will be divided into three main parts. The first part discusses about how genetic algorithm and also heuristic search can solve scheduling problem. Some related works are also included. The second part will be proposed the architecture design of the system. The third part will be shown some experiments and discussion after implementing the system. Chapter will be closed by the conclusion and also some suggestions to improve the system.

2. Principle of genetic algorithm and time tabling

2.1 Principle of genetic algorithm

Genetic Algorithms (GA) are powerful general purpose optimization tools which model the principles of evolution (Davis L. 91). They are often capable of finding globally optimal solutions even in the most complex of search spaces. They operate on a population of coded solutions which are selected according to their quality then used as the basis for a new generation of solutions found by combining (crossover) or altering (mutating) current individuals. Traditionally, the search mechanism has been domain independent, that is to say the crossover and mutation operators have no knowledge of what a good solution would be (Bruns93)(Burke et al.94).

The working principle of a canonical GA is illustrated in Fig. 1. The major steps involved are the generation of a population of solutions, finding the objective function and fitness function and the application of genetic operators. These aspects are described briefly in the subsection below.

An important characteristic of genetic algorithm is the coding of variables that describes the problem. The most common coding method is to transform the variables to a binary string or vector. This initial population formulation process is critical. This step is also recognized as encoding process.

```
formulate initial population
randomly initialize population
repeat
    evaluate objective function
    apply genetic operators
        reproduction
        crossover
        mutation
until stopping criteria
```

Fig. 1. The Working Principle of a Simple Genetic Algorithm

GA processes a number of solutions simultaneously. Hence, in the first step a population having P chromosomes called individuals is generated by pseudo random generators whose individuals represent a feasible solution. This is a representation of solution vector in a solution space and is called initial solution. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space.

In the next step, individual members, chromosomes of the population represented by a string are evaluated to find the objective function value. This is exclusively problem specification. The objective function is mapped into a fitness function that computes a fitness value for each chromosome. This is followed by the application of GA operators.

Reproduction or selection is usually the first operator applied on a population. It is an operator that makes more copies of better chromosomes in a new population. Thus, in reproduction operation, the process of natural selection causes those chromosomes that encode successful structures to produce copies more frequently. To sustain the generation of a new population, the reproduction of the chromosomes in the current population is necessary. For better chromosomes, these should be generated from the fittest chromosomes of the previous population.

There exist a number of reproduction operators in GA literature, but the essential idea in all of them is that the above average fitness value of strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner.

A crossover operator is used to recombine two chromosomes to get a better one. In the crossover operation, recombination process creates different chromosomes in the successive generations by combining material from two chromosomes of the previous generation. In reproduction, good chromosomes in a population are probabilistically assigned a larger number of copies and a mating pool is formed. It is important to note that no new chromosomes are usually formed in the reproduction phase. In the crossover operator, new chromosomes are created by exchanging information among strings of the mating pool.

The two chromosomes participating in the crossover operation are known as parent chromosomes and the resulting ones are known as children chromosomes. It is intuitive from this construction that good sub-strings from parent chromosomes can be combined to form a better child chromosome, if an appropriate site is chosen. With a random site, the children chromosomes produced may or may not have a combination of good sub-strings

from parent chromosomes, depending on whether or not the crossing site falls in the appropriate place. But this is not a matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover.

It is clear from this discussion that the effect of crossover may be detrimental or beneficial. Thus, in order to preserve some of the good chromosomes that are already present in the mating pool, all chromosomes in the mating pool are not used in crossover. When a crossover probability, defined here as p_c is used, only 100 multiplied by p_c per cent chromosomes in the population are used in the crossover operation and 100 multiplied by $(1-p_c)$ per cent of the population remains as they are in the current population. A crossover operator is mainly responsible for the search of new chromosomes even though mutation operator is also used for this purpose sparingly.

Many crossover operators exist in the GA literature (Zhao, 2007). One site crossover and two site crossover are the most common ones adopted. In most crossover operators, two strings are picked from the mating pool at random and some portion of the strings is exchanged between the strings. Crossover operation is done at string level by randomly selecting two strings for crossover operations.

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes to be different from those of their parent. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level. When the bits are being copied from the current string to the new chromosomes, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability p_m. The need for mutation is to create a point in the neighborhood of the current point. The mutation is also used to maintain diversity in the population.

These three operators are simple and straightforward. The reproduction operator selects good chromosomes and the crossover operator recombines good sub-strings from good strings together, hopefully, to create a better sub-string chromosome. The mutation operator alters a string locally expecting a better chromosome. Even though none of these claims are guaranteed and/or tested while creating a chromosome, it is expected that if bad chromosomes are created they will be eliminated by the reproduction operator in the next generation and if good chromosomes are created, they will be increasingly emphasized.

Further insight into these operators, different ways of implementations and some mathematical foundations of genetic algorithms can be obtained from GA literature (Zhao, 2007). Application of these operators on the current population creates a new population. This new population is used to generate subsequent populations and so on, yielding solutions that are closer to the optimum solution. The values of the objective function of the chromosomes of the new population are again determined by decoding the strings. These values express the fitness of the solutions of the new generations. This completes one cycle of genetic algorithm called a generation. In each generation if the solution is improved, it is stored as the best solution. This is repeated till convergence as depicted in Figure 2.

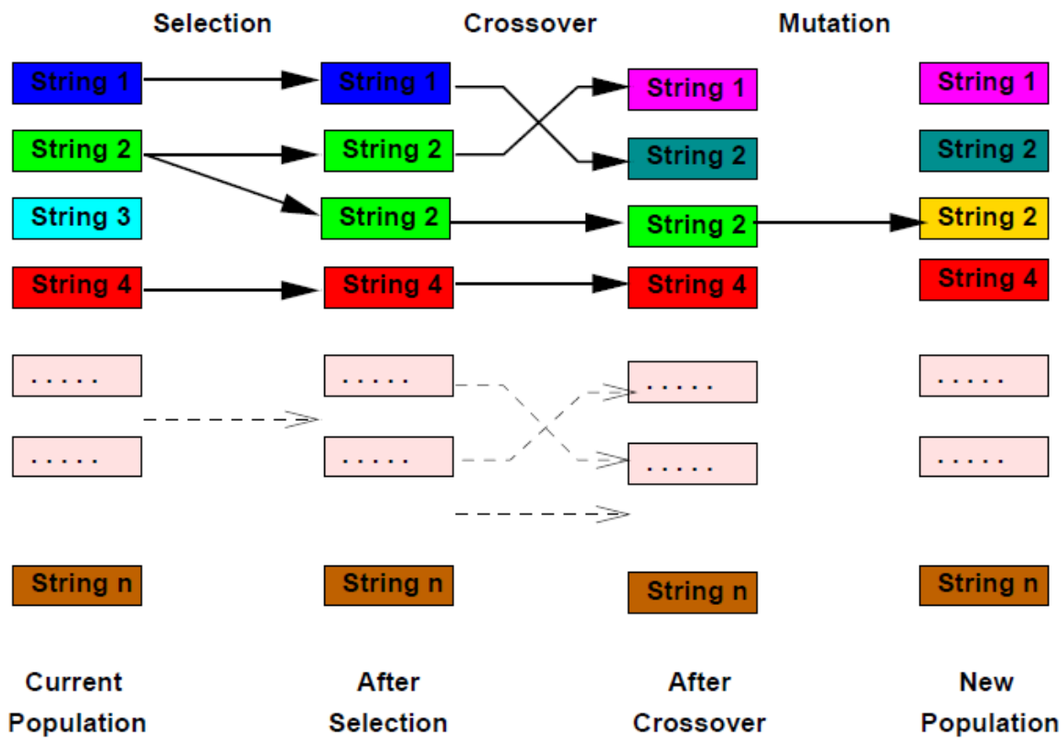


Fig. 2. The basic GA operations

Encoding technique used in this research is permutation encoding. In this technique, chromosomes are coded in the form of integers. Each integer called gene is uniquely assigned to a certain course taught by a lecture. Sequence of integer representing genes in a chromosome, determines sequence of courses to be scheduled (Thanh, 2007). For example a five-gene chromosome represented by 4 2 3 1 5 means that course represented by integer 4 will be scheduled first. Later, it will be followed by course represented by others integer consecutively. Strings length is a total number of courses to be scheduled in a course group. If there are four courses to be scheduled and each of them will be scheduled twice a week, then chromosome length is 8.

Selection is conducted based on truncation selection. Chromosomes are sorted according to their fitness value from the biggest to the smallest. Some chromosomes, started from the smallest fitness strings, will be replaced by new ones (Zhao, 2007). A new chromosome is obtained by reversing the position of all bits in an old chromosome. Unlike other selection methods, the truncation selection does not copy the better chromosome to the population but create a new one. Number of chromosomes to be replaced is obtained by multiplying number of all chromosomes in population with probability of selection. For example, Table 1 contains five-gene sorted chromosomes.

If probability of selection is 0.4, it means that the number of old chromosomes that must be replaced by new ones is 5*0.4 = 2. Then, the position of genes in the last two chromosomes will be reversed. Chromosomes 1 2 5 3 4 will be replaced by 4 3 5 2 1, whereas chromosomes 5 2 3 4 1 will be replaced by 1 4 3 2 5.

Chromosome	Fitness
1 4 5 2 3	1
3 5 2 1 4	0.95
2 4 5 1 3	0.7
1 2 5 3 4	0.5
5 2 3 4 1	0.3

Table 1. Chromosomes before selection

Cycle crossover is applied in this research. The idea behind this method is finding the genes cycle between two parents. Genes that are included in the cycle will stay, while the others will be swapped between the two parents, in order to form two children (Lukas et al, 2005). For example, the two parents used as shown in Figure 3 are 1 5 3 4 2 and 3 4 2 5 1. Genes cycle from those two parents is 1 3 2 1. Then, gene 1, 2 and 3 will stay in that position, while gene 4 and 5 will be swapped. It can be seen that gene 5 of first parent is swapped with gene 4 of second parent, and gene 4 of first parent is swapped with gene 5 of second parent. Therefore chromosomes of the two children are 1 4 3 5 2 and 3 5 2 4 1. Number of crossover is calculated by multiplying number of populations with probability of crossover. It represents how many of chromosomes in population will be crossovered.

In mutation phase, reciprocal exchange mutation is used. Each mutation using this method causes two genes mutated at the same time. First step of this method is determining two gene positions randomly. Then, genes in those positions are swapped (Lukas et al, 2005). For example in chromosome 3 2 5 4 1, the two gene positions chosen are the second and the third. Then, genes in those positions that are gene 2 and 5 are swapped. Chromosome obtained after mutation is 3 5 2 4 1. Number of mutation in population is counted by multiplying chromosome length, number of populations and probability of mutation. Number of crossover and number of mutation must be an even number, because in each crossover, two chromosomes are combined, while in each mutation, two genes are swapped.

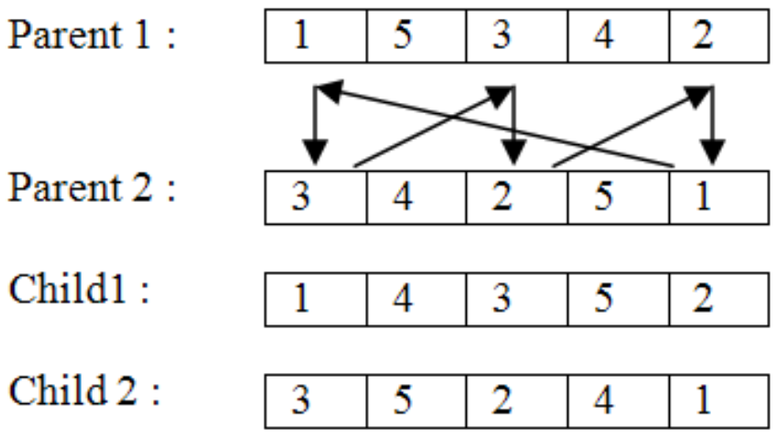


Fig. 3. Cycle Crossover

Fitness value is used to determine how good a chromosome is. It indicates what the objective of the chromosome is. Usually the value is grether than or equal to zero but less or equal to 1. In this research, the objective is to maximize the number of units being able to scheduled divided by total units. So, the more successfully units scheduled courses are, the bigger the fitness value of a chromosome is. Each course has a certain unit. This unit means how many hour students have to spend their time for that course. If a credit of a course is 2, it means that the scheduled course is two hours a week in the timetable consecutively.

2.1 Time tabling

A timetable is essentially a schedule which must suit a number of constraints. Constraints are almost universally employed by people dealing with timetabling problems (Burke and Ross, 1996). Constraints, in turn, are almost universally broken into two categories: soft and hard constraints. Hard constraints are constraints, of which, in any working timetable, there will be no breaches. For example, a lecturer cannot be in two places at once (Erben and Keppler, 1995; Rich 1995). Soft constraints are constraints which may be broken, but of which breaches must be minimized. For example, classes should be booked close to the home department of that class (Erben and Keppler, 1995). In addition to constraints, there are a number of exceptions which must be taken into consideration when constructing an Automated Timetabling system.

In this research, the hard constraints are classrooms must not be double booked, every class must be scheduled exactly once, classes of students must not have two bookings simultaneously, a classroom must be large enough to hold each class booked to it, lecturers must not be double booked, a lecturer must not be booked when he/she is unavailable. Some classes require particular rooms; some classes need to be held consecutively. Whereas the soft constraints are some lecturers have preferred hours to be scheduled, most students do not wish to have empty periods in their timetables, the distance a student walks should be minimized, classes should be distributed evenly over the week, classrooms should be booked close to the home department of that class, classrooms should not be booked which are much larger than the size of the class. In addition, the only exception constraint to be considered is a part time lecturer should be scheduled not more then 6 credit units whereas a full time is 12 credit units.

3. The architecture design of the system

Heuristic search is applied in this research. It uses a 2D matrix called target matrix. This matrix is used to find suitable time slots for scheduling courses (Thanh, 2007). There are six sets applied in this target matrix. They are course code set $M=\{m_1,m_2,\ldots\}$, type course class set $T=\{t_0,t_1,t_2,\ldots\}$, lecturer code set $L=\{l_1,l_2,\ldots\}$, class name set $C=\{c_1,c_2,\ldots\}$, day set $D=\{d_1,d_2,\ldots\}$ and hour set $H=\{h_1,h_2,\ldots\}$. All index set start with one. Only type course class set T has member index zero, t_0. It indicates only for the case that for a certain course of some parallel classes are merged into one class.

There are close relations among course code set, type course class set and also lecture code set. Suppose m_1,t_1,l_1 is one of the relations, it indicates that course code m_1 is taught by lecture code l_1 in type course class code t_1. In addition, c_1,d_1,h_1 is also another relation, it

means that at day d_1 on hour h_1 class name c_1 is assigned. If m_1,t_1,l_1 is linked with c_1,d_1,h_1, it means that course code m_1 is taught by lecture code l_1 in type course class code t_1 be scheduled at day d_1 on hour h_1 with class name c_1.

All pairs of course code, type course class and lecture code have to be connected with all pairs of class name, day and hour. This connection is tabulated in a matrix called target matrix. Table 2 is an example of target matrix. Each cell v_{ij} in which i is row and j is column, in target matrix, $V = \{v_{ij}\}$, has three different values, those are :

1. v_{ij} = 0 means that certain lecturer time slot represented by this cell is available to be scheduled.
2. v_{ij} = -1 means that certain lecturer time slot represented by this cell is not available to be scheduled.
3. v_{ij} = 1 means that certain lecturer time slot represented by this cell has already been scheduled.

	m_1,t_0,l_1	m_2,t_1,l_1	m_3,t_2,l_2	...
c_1,d_1,h_1	1	-1	-1	...
c_1,d_1,h_2	1	-1	-1	...
c_1,d_2,h_1	-1	1	-1	...
c_1,d_2,h_2	-1	1	-1	...
c_2,d_1,h_1	1	-1	-1	...
c_2,d_1,h_2	1	-1	-1	...
c_2,d_2,h_1	0	0	0	...
c_2,d_2,h_2	-1	-1	1	...
...	...	...		
# of units scheduled	4	2	1	...

Table 2. Example of target matrix

Number of rows needed is equal to number of class names multiplied by number of lecture days in a week multiplied by number of lecture hours in a day. From Table 2, there are two names of class, c_1 and c_2, are scheduled in the same day and time that is d_1,h_1 and d_1,h_2 for course m_1 taught by l_1. It is possible because the type of class is t_0. It means that it is a merge class of n_1 and n_2. Number of units scheduled of that column is 4. It represents a number of hours has been allocated.

There are six functions that are applied to each cell in target matrix. Those functions are f_m , f_t , f_l , f_c , f_d and f_h, each of which is used to get information about course code, type course class, lecturer code, class name, day and hour respectively. As an example, the value of each function to cell v_{11} are $f_m(v_{11})=m_1$, $f_t(v_{11})=t_0$, $f_l(v_{11})=l_1$, $f_c(v_{11})= c_1$, $f_d(v_{11})= d_1$, and $f_h(v_{11})= h_1$. All of these functions are used to create target matrix.

There are some rules need to be considered in order to fill the target matrix:

1. If $t_x \neq t_0$ and certain course respectively should be scheduled in class name i^{th} then all rows $c_j \neq i$ in that column should not be filled by 1. In addition, if that course can be scheduled at the certain day and hour at that column then that cell at rows c_i is assigned as 1 and others consecutive cells in the same day, as many as unit course of that column, is also assigned 1.
2. If $t_x = t_0$ means that certain course respectively should be scheduled in c_i and c_j class names. Therefore if and only if cell $c_i,d_a,h_b = 1$ then cell $c_j,d_a,h_b = 1$ at that column.
3. For each row, there is only maximum a cell that is equal to 1.The others must be -1 or 0.
4. Course on a column is said to be successfully scheduled only when unit course at that column is a factor of total all 1's cells in that column.
5. If there is a cell in a column of a row c_x, d_i, h_j is equal to 1 then for all row c_y, d_i, h_j have to be set -1 for $c_x \neq c_y$ at that column.

For example, there are 3 courses and 4 lecturers to be scheduled within 2 class names, A and B, 2 days lecture a week and 3 hours lecture a day. There are also two type course classes, namely lecture and lab. It means that $n(M)$= 3, $n(T)$=3, $n(L)$=4, $n(C)$=2, $n(D)$=2, $n(H)$=3. m_1 is course code opened for mixed class name c_1 and c_2 and lectured by l_1, m_2 is course code opened for both class name c_1 and c_2 but their were taught by the some lecturer, l_2 with type course class t_1 as lecture. m_3 is course code also opened for both class, type course class as lab, but with difference lecturer, l_3 and l_4. If course credit of m_1, m_2 and m_3 are 2, 1 and 2 credits and knowing the lecturer's availabilities time, we can produce the initial target matrix, in Table 3. From that target matrix, it can be determined that lecture code l_2 and l_4 are not available for d_1,h_1 and d_2,h_3, while other lecturers are available at any time. These information and other constraints are inputted into the system and saved into databases.

	m_1,t_0,l_1	m_2,t_1,l_2	m_3,t_2,l_3	m_3,t_2,l_4
c_1,d_1,h_1	0	-1	0	0
c_1,d_1,h_2	0	0	0	0
c_1,d_1,h_3	0	0	0	0
c_1,d_2,h_1	0	0	0	0
c_1,d_2,h_2	0	0	0	0
c_1,d_2,h_3	0	0	0	-1
c_2,d_1,h_1	0	-1	0	0
c_2,d_1,h_2	0	0	0	0
c_2,d_1,h_3	0	0	0	0
c_2,d_2,h_1	0	0	0	0
c_2,d_2,h_2	0	0	0	0
c_2,d_2,h_3	0	0	0	-1
# units	0	0	0	0

Table 3. Example of initial target matrix

Suppose that sequence of courses represented by generated chromosome [1 3 2 4] represents $m_{1,}t_0$,l_1 ; $m_{3,}t_2$,l_3 ; $m_{2,}t_1$,l_2 and $m_{3,}t_2$,l_4 then, the first course to be scheduled is gene chromosome code $m_{1,}t_0$,l_1 and the last is $m_{3,}t_2$,l_4. The result of these is shown in Table 4. It can be seen that $m_{1,}t_0$,l_1 is successfully scheduled on (c_1, d_1, h_1), (c_1, d_1, h_2) (c_2, d_1, h_1) and (c_2, d_1, h_2). Therefore other columns of those rows are set to -1. $M_{3,}t_2$,l_3 and $m_{2,}t_1$,l_2 are able to be placed but $m_{3,}t_2$,l_4 is failed.

From Table 4, it can be said that the chromosome [1 3 2 4] is able to allocated 8 units out of 10 units for that two classes c_1 and c_2 for 3 courses. The value of 8 over 10 which is 0.8 is called as fitness of that chromosome. If the chromosome is [1 4 3 2] then the fitness is 1. The fitness of a chromosome represents the time table objective. In this case, the objective is to maximize the number of units to be scheduled for each timetable. If number of units scheduled for k^{th} column is $s(k)$, p is a maximum column in a target matrix, $u(j)$ means unit number of j^{th} course to be scheduled for each class name, and t is total courses of each class name, then fitness of a chromosome is defined by (1)

$$fitness = \frac{\sum_{k=1}^{p} s(k)}{n(C) \cdot \sum_{j=1}^{t} u(j)} \tag{1}$$

	m_1, t_0, l_1	m_2, t_1, l_2	m_3, t_2, l_3	m_3, t_2, l_4
c_1,d_1,h_1	1	-1	-1	-1
c_1,d_1,h_2	1	-1	-1	-1
c_1,d_1,h_3	-1	1	-1	-1
c_1,d_2,h_1	-1	-1	1	-1
c_1,d_2,h_2	-1	-1	1	-1
c_1,d_2,h_3	0	0	0	-1
c_2,d_1,h_1	1	-1	-1	-1
c_2,d_1,h_2	1	-1	-1	-1
c_2,d_1,h_3	0	-1	0	0
c_2,d_2,h_1	-1	1	-1	-1
c_2,d_2,h_2	0	0	-1	0
c_2,d_2,h_3	0	0	0	-1
# units	4	2	2	0

Table 4. Example of target matrix after heuristic search

University time table consists of many timetables. Since one degree of each department is designed for four years studies, then each department has at least four time tables for every

semester. That is one time table for every batch. No matter is how big a batch in one departement, it has only one time table. It only impacts to the processing time. The bigger the batch is the longer the processing time to produce the time table. It is because not only the number of rows but also the number of columns in the target matrix will be larger.

4. Experiment result

Some experiments are performed to ensure how good the system is. There are 4 course groups from year 1 to year 4. In experiments, we would like making time table for odd semester data 2008 - 2009. Certain number of courses and lecturers to be scheduled in each course group are inputted. For example, in course group year 1 there are 7 courses and 10 lecturers with 2 type course classes (lecture and lab), 2 class names (A and B), 5 days lecture a week and 10 hours lecture a day. Relation among all sets are represented in Table 5.

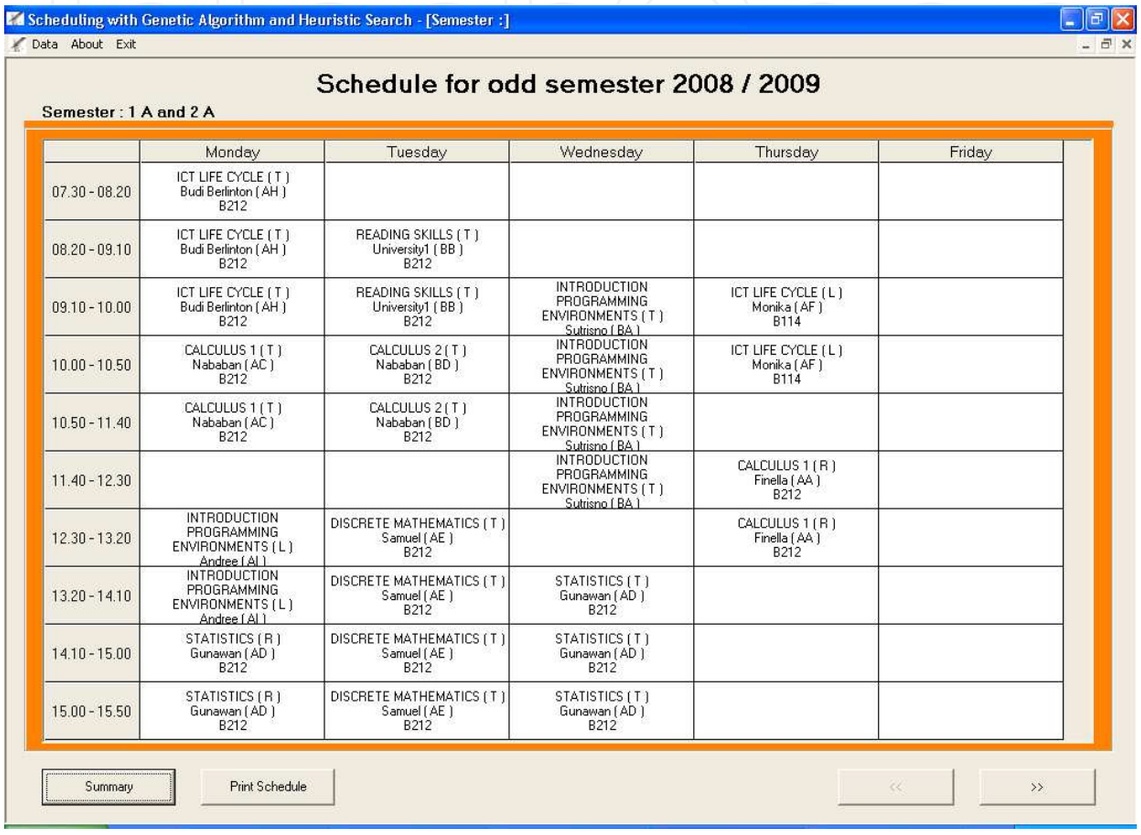
From Table 5, it can be concluded that $n(M)$= 7, $n(T)$=3, $n(L)$=9, $n(C)$=2, $n(D)$=5, $n(H)$=10. All genes of the chromosome are m_1,t_0,l_1, m_2,t_0,l_2, m_3,t_0,l_2, m_4,t_0,l_3, m_5,t_0,l_6, m_6,t_0,l_5, m_7,t_1,l_6, m_7,t_1,l_7, m_1,t_2,l_8, m_2,t_2,l_9, m_3,t_2,l_9, m_4,t_2,l_{10}, m_6,t_2,l_6. It means there are 13 columns and 100 rows in target matrix. After receiving initial data, such as what is the restricted day and time for a certain lecturer, what room can be used and also the capacity of that room. The system runs with 10 chromosomes in a population and 10 generations are set in the experiment, without considering probability of selection, crossover and mutation, the maximum best fitness value, that is 1, can be achieved. It means that all courses can be scheduled accordingly. Therefore, number of populations and generations does not need to be set high. It means less time is needed to make a schedule. One of the timetables of the course group year 1 is shown in Figure 4.

Course Name	Lecture			Lab		
	A	B	units	A	B	units
ICT (m_1)	Budi Berlinton (l_1)		3	Monika (l_8)	Monika (l_8)	2
Calculus 1 (m_2)	Nababan (l_2)		2	Finela (l_9)	Finela (l_9)	2
Calculus 2 (m_3)	Nababan (l_2)		2			
IPE (m_4)	Sutrisno (l_3)		4	Andree (l_{10})	Andree (l_{10})	2
Discrete Math (m_5)	Samuel (l_4)		4			
Statistics (m_6)	Gunawan (l_5)		3	Gunawan (l_5)		2
Reading Skills (m_7)	Univ (l_6)	Univ (l_7)	2			

Table 5. Relations among the 6 sets

The time table where the class meeting will take place should also be defined. From figure 4, it is clear that every class meeting has their room. Budi berlinton teaches course ICT on

Monday at 7.30 to 10.00 in room B212. The system can define the available room for that class by looking room table in the database. It is easy to search the room by comparing the room capacity, the location etc, from the requirement of that class. After a room is assigned, status of that room is not available any more of that day and hour.



	Monday	Tuesday	Wednesday	Thursday	Friday
07.30 - 08.20	ICT LIFE CYCLE (T) Budi Berlinton (AH) B212				
08.20 - 09.10	ICT LIFE CYCLE (T) Budi Berlinton (AH) B212	READING SKILLS (T) University1 (BB) B212			
09.10 - 10.00	ICT LIFE CYCLE (T) Budi Berlinton (AH) B212	READING SKILLS (T) University1 (BB) B212	INTRODUCTION PROGRAMMING ENVIRONMENTS (T) Sutrisno (BA)	ICT LIFE CYCLE (L) Monika (AF) B114	
10.00 - 10.50	CALCULUS 1 (T) Nababan (AC) B212	CALCULUS 2 (T) Nababan (BD) B212	INTRODUCTION PROGRAMMING ENVIRONMENTS (T) Sutrisno (BA)	ICT LIFE CYCLE (L) Monika (AF) B114	
10.50 - 11.40	CALCULUS 1 (T) Nababan (AC) B212	CALCULUS 2 (T) Nababan (BD) B212	INTRODUCTION PROGRAMMING ENVIRONMENTS (T) Sutrisno (BA)		
11.40 - 12.30			INTRODUCTION PROGRAMMING ENVIRONMENTS (T) Sutrisno (BA)	CALCULUS 1 (R) Finella (AA) B212	
12.30 - 13.20	INTRODUCTION PROGRAMMING ENVIRONMENTS (L) Andree (AI)	DISCRETE MATHEMATICS (T) Samuel (AE) B212		CALCULUS 1 (R) Finella (AA) B212	
13.20 - 14.10	INTRODUCTION PROGRAMMING ENVIRONMENTS (L) Andree (AI)	DISCRETE MATHEMATICS (T) Samuel (AE) B212	STATISTICS (T) Gunawan (AD) B212		
14.10 - 15.00	STATISTICS (R) Gunawan (AD) B212	DISCRETE MATHEMATICS (T) Samuel (AE) B212	STATISTICS (T) Gunawan (AD) B212		
15.00 - 15.50	STATISTICS (R) Gunawan (AD) B212	DISCRETE MATHEMATICS (T) Samuel (AE) B212	STATISTICS (T) Gunawan (AD) B212		

Fig. 4. One example of time table for the first semester.

5. Conclusions and further research

The proposed genetic algorithm and heuristic search are able to solve timetable problem. Although, room has not been included in target matrix, system is able to determine which room is used to a certain cell in time table. However, if the room is one of the critical factor, it should be included in the target matrix. If it happens then it is more likely to create three dimension target matrix instead of adding number of column.

There are some limitations of this research. Firstly, every parallel class of a course group which is represented in one target matrix, has to be scheduled to every course defined in that time table. Secondly, one lecture course can be scheduled only if there is available space consecutively at target matrix at least as much as number of units of the course. If that

course has to be split into two segments then the name of that course should be differentiated. It could be assumed as two courses. In the experiment, it is showed that calculus is divided into calculus 1 and calculus 2. Thirdly, the objective of the system is to maximize the number of successful units being able to scheduled, otherwise it should be defined accordingly.

6. References

Burke E.K., Elliman D.G. and Weare R.F. (1993a) "A University Timetabling System Based on Graph Coloring and Constraint Manipulation", *Journal of Research on Computing in Education*. Vol. 26. issue 4

Burke E.K., Elliman D.G. and Weare R.F. (1993b) "Automated Scheduling of University Exams", *Proceedings of I.E.E. Colloquium on "Resource Scheduling for Large Scale Planning Systems"*, Digest No. 1993/144

Burke E.K., Elliman D.G. and Weare R.F. (1994) "A Genetic Algorithm for University Timetabling", AISB Workshop on Evolutionary Computing, Leeds.

Burke E and Ross P (Eds) (1996): *Lecture Notes in Computer Science 1153 Practice and Theory of Automated Timetabling First International Conference, Edinburgh, U.K., August/September 1995, Selected Papers.* New York: Springer-Verlag Berlin Heidelberg.

Davis L. (1991) "Handbook of Genetic Algorithms" Van Nostrand Reinhold

Erben W and Keppler J (1995): A Genetic Algorithm Solving a Weekly Course- Timetabling Problem. In Burke E and Ross P (Eds): *Lecture Notes in Computer Science 1153 Practice and Theory of Automated Timetabling First International Conference, Edinburgh, U.K., August/September 1995, Selected Papers.* New York: Springer-Verlag Berlin Heidelberg. pp 198-211.

Enmund Burke, David Ellimand and Rupert Weare. (1994). "*A Genetic Algorithm Based University Timetabling System*", August 2011, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.2659

Joshua Poh-Onn Fan, Graham K. Winley. (2008). *A Heuristic Search Algorithm for Flow-Shop Scheduling"*, August 2011, http://www.informatica.si/PDF/32-4/26_Fan - A Heuristic Search Algorithm for Flow-Shop Scheduling.pdf

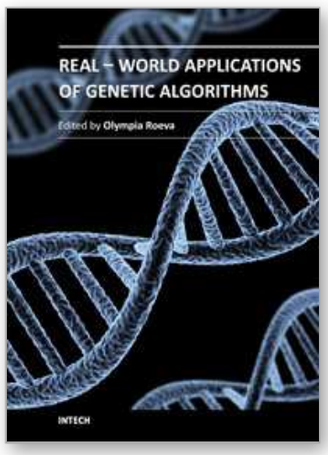
N.D. Thanh, "Solving Timetabling Problem Using Genetic and Heuristic Algorithms", Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, hal. 472-477, 2007.

Negnevitsky, Michael. (2005). *Artificial Intelligence, A Guide to Intelligence System (2nd)*, Addison Wesley, pp. 222-245, IBN 0-321-20466-2, Harlow, England

Q. Zhao, An Introduction to Genetic Algorithms, 2007.

Rich DC (1995): A Smart Genetic Algorithm for University Timetabling. In Burke E and Ross P (Eds): *Lecture Notes in Computer Science 1153 Practice and Theory of Automated Timetabling First International Conference, Edinburgh, U.K., August/September 1995, Selected Papers.* New York: Springer-Verlag Berlin Heidelberg. pp 181-197.

S. Lukas, P. Yugopuspito and H. Asali, "Solving assignment problem by genetic algorithms using Cycle Crossover", *Universitas Pelita Harapan Computer Science Journal,* Vol. 3, No. 2, Mei 2005, pp. 87-93.



Real-World Applications of Genetic Algorithms
Edited by Dr. Olympia Roeva

ISBN 978-953-51-0146-8
Hard cover, 376 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

The book addresses some of the most recent issues, with the theoretical and methodological aspects, of evolutionary multi-objective optimization problems and the various design challenges using different hybrid intelligent approaches. Multi-objective optimization has been available for about two decades, and its application in real-world problems is continuously increasing. Furthermore, many applications function more effectively using a hybrid systems approach. The book presents hybrid techniques based on Artificial Neural Network, Fuzzy Sets, Automata Theory, other metaheuristic or classical algorithms, etc. The book examines various examples of algorithms in different real-world application domains as graph growing problem, speech synthesis, traveling salesman problem, scheduling problems, antenna design, genes design, modeling of chemical and biochemical processes etc.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen