

Genetic Algorithm for Lecturing Schedule Optimization (Case Study: University of Boyolali)

David Kristiadi*¹, Rudy Hartanto²

¹Sekolah Tinggi Multi Media, Jl. Magelang Km.6 Yogyakarta

²Departemen Teknik Elektro dan Teknologi Informasi, Fakultas Teknik UGM, Yogyakarta

e-mail: *¹dv.davk@gmail.com, ²rudy@ugm.ac.id

Abstrak

Penjadwalan merupakan masalah klasik di dalam perkuliahan. Ruang, dosen, waktu kuliah dan ketentuan-ketentuan penjadwalan harus dikelola dengan baik untuk mencapai penjadwalan kuliah yang optimal. Permasalahan penjadwalan kuliah juga dialami oleh Universitas Boyolali (UBY). Penyelesaian dari permasalahan tersebut didekati dengan membangun sebuah library dengan konsep Algoritma Genetika (AG). AG adalah sebuah metode komputasi yang terinspirasi oleh proses seleksi alam. Library yang dibangun tersusun oleh beberapa operator seperti Turnamen Selection, Uniform Crossover, Weak Parent Replacement dan dua metode mutasi yaitu Interchanging Mutation dan Violated Directed Mutation (VDM). Kedua metode mutasi dibandingkan untuk mendapatkan metode mutasi yang lebih baik. Library juga dirancang agar mampu mendefinisikan custom constraint atau ketentuan penjadwalan yang belum diakomodasi di dalam library tanpa modifikasi inti program. Hasil ujicoba menunjukkan bahwa mutasi dengan VDM lebih menjanjikan solusi yang optimal daripada menggunakan Interchanging Mutation. Berkaitan kasus UBY, solusi optimal (nilai fitness=1) diperoleh dalam waktu 12 menit 41 detik dengan penambahan 6 ruang kelas baru dan menonaktifkan 2 konstrain yaitu perkuliahan dimulai jam 14.00 kecuali untuk Program Studi Hukum semester 3 kelas pagi dan peserta kuliah tidak boleh melebihi kapasitas.

Kata kunci—Algoritma genetika, Violated Directed Mutation, VDM vs Interchanging Mutation

Abstract

Scheduling is a classic problem in lecturing. Rooms, lecturers, times and scheduling constraints must be managed well to get an optimal schedule. University of Boyolali (UBY) also encounter the same scheduling problems. The problem was tried to be solved by building a library based on Genetic Algorithm (GA). GA is a computation method which inspired by natural selection. The computation consists of some operators i.e. Tournament Selection, Uniform Crossover, Weak Parent Replacement and two mutation operators (Interchanging Mutation and Violated Directed Mutation (VDM)). The two mutation method are compared to find which better mutation operator. The library was planned to have a capability to define custom constraints (scheduling requirements that were not accommodated by the library) without core program modifications. The test results show that VDM is more promising for optimal solutions than Interchanging Mutation. In UBY cases, optimal solution (fitness value=1) is reached in 12 minutes 41 second with adding 6 new room and inactivated 2 constraint i.e. lecturing begins at 14.00 except for 3rd semester of science law study program with morning class and lecturing participants must not over classroom capacity.

Keywords—Genetic algorithm, Violated Directed Mutation, VDM vs Interchanging Mutation

1. INTRODUCTION

University of Boyolali (UBY) is one of some colleges in Boyolali regency. Since established on February 11th, 2008, UBY has 7 active study program and registered on Forlap DIKTI [1]. UBY has also the same classical problem in scheduling like other colleges. The problem is the timetable is not optimal. When a lecturing schedule fulfills all scheduling constraint, it is called as an optimal schedule. Example of scheduling constraints are no crash schedule, student and lecturer learning activity are distributed well in a timetable period. Insufficient human resources, rooms and time are the problems for schedule administrator. They must thorough making timetable in order to get an optimal lecturing schedule.

Some scheduling application with many computation methods has been researched and built. Genetic algorithm (GA) is one of the computation methods that used in scheduling application. GA runs a looping operation (selection, crossover, mutation, and replacement) until looping stop condition triggered. Using GA in scheduling was reported that it can produce an optimal schedule [2–12]. GA can be used in a complete operation step [2, 6, 7, 9, 11, 12], in a modification operation step e.g. eliminate crossover [3, 8] or in combination operation step with another computation method [5, 10]. Scheduling in university still has a problem in find general and effective solution that depends on schedule diversification problems, constraint variations and specific terms appropriate to characteristics of the university [13]. That why, it is necessary to build a concept of a library that accommodates schedule diversifications problems, constraint variations and specific terms appropriate to characteristics of the university.

2. METHODS

2.1 UBY Scheduling Data

Data and fact of scheduling that collected, UBY lecturing schedule has some constraints. The constraints are (C-1) No crash among lecturers, students and rooms schedule. (C-2) 12.00 – 13.00 is a break/free time; (C-3) Lecturing begins at 14.00 except for 3rd semester of science law study program with morning class; (C-4) Science law study program with morning class must finish lecturing before 12.00 each day; (C-5) Every classroom can be used by another study programs; (C-6) Lecturing participants must not over classroom capacity; (C-7) The activity load per day of students and lecturers are flexible (it can be ignored).

The other data of UBY scheduling at First Semester of 2017/2018 is shown in Table 1. There is a problem here e.i. a few of classroom (it is not comparable with the number of room users) and time of lecturing mostly begin at 14.00.

Table 1. The summary of UBY scheduling data at First Semester of 2017/2018

Data	Total	Notes
Lecturer	65 lecturers	Around 65 persons
Study Program	7	Totally 16 study programs and only 7 study programs are active
Courses	166 Courses	At the first, totally there are 183 courses. After applying filter appropriate to the campus policy that the same course with the same lecturer should be joined in one class, so finally there are 166 courses.
Day of lecturing	6 days	Monday – Saturday
Total credits (SKS) at first semester of 2017/2018	479 SKS	It consists of 24 SKS for the morning class and 455 SKS for the afternoon class. After applying filter appropriate to the campus policy campus policy that the same course with the same lecturer should be joined in one class, totally there are 438 SKS (414 SKS at afternoon and 24 SKS at morning).
Time	11 hours	Start from 08.00 – 20.00, break time at 12.00–13.00; The afternoon class begins at 14.00. The morning class begins at 08.00 only for the 3 rd semester Law Science study program.
Classroom	8 Classroom	There are 10 rooms, but 4 rooms joined to be 2 rooms.
Lecturing Slot (LS)	528 LS	LS total (Lecturing and break time) = 6 days x 8 rooms x 12 hours = 576 LS Lecturing slot (lecturing only) = 6 days x 8 rooms x 11 hours = 528 LS The morning class LS (Begin at 8-12)= 6 days x 8 rooms x 4 hours = 192 LS The afternoon class LS (Begin at 14-20)= 6 day x 8 rooms x 6 hours = 288 LS

2.2 System Design

The output of this research is a javascript library for lecturing schedule optimization using Genetic Algorithm (OPK-GA library). This library should be developed more to be end-user scheduling application. The OPK Library plan is shown in Figure 1. The library needs 2 types of input data. They are raw data of lecturing schedule and GA parameters. The outputs of the library are the lecturing schedule or the computation results. The types of inputs and outputs of the library are JSON (Javascript Object Notation). Many program languages can support JSON data types.

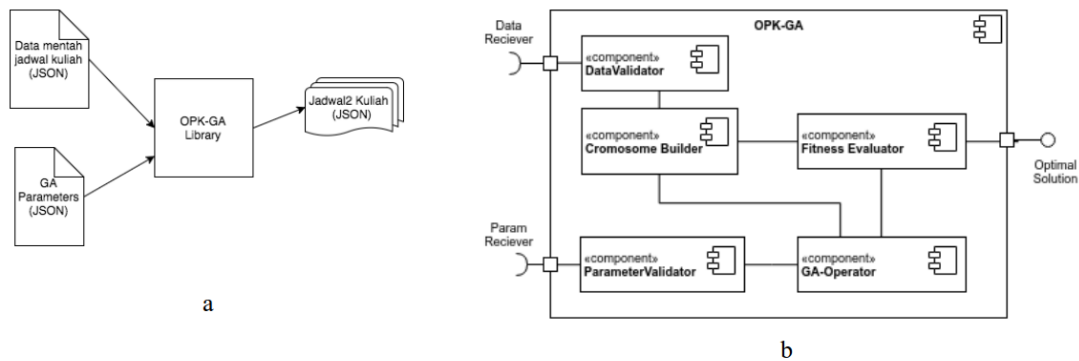


Figure 1 The plan of OPK-GA;

(a) Conceptual diagram of OPK-GA; (b) Component Diagram of OPK-GA

OPK-GA consists of some components as shown in Figure 1 (b). ParameterValidator component is a data type and schema validator for GA parameter input. GA parameter consists of some configurations to control the computation process. DataValidator component is a data type and schema validator for raw data of lecturing schedule input. A validated data will be inputted to ChromosomeBuilder component. After that, ChromosomeBuilder will construct several individuals as the initial population. The initial population will be evaluated by FitnessEvaluator component. After that, the population will be inputted to GA-Operator component to run GA Operation process. GA Operation process consists of selection, crossover, mutation, and replacement.

2.3 Input and Output Design

Input and output system is shown in Figure 2. Input consists of raw data of lecturing schedule and GA parameters. The output system is lecturing schedules. Raw data of lecturing schedule, as shown in Figure 2 (a), consists of (1) days as the name of lecturing days; (2) time as fragmentation of lecturing time in one day; (3) rooms as the name of classroom; (4) roomProps contains 2 data (room capacity and room ownership); (5) courses consists of lecturer and courses plotting, it should contain some informations such as prodi (study program name), course (course name), sks (credits), students (the number of participants), required, lecturer (lecturer name), and smt (the semester where the course should be taken by student); (6) constraints consist of room constraints, lecturer constraints, fixedCourses (some courses with fixed schedule), and others (the others constraints that are not included in the system by default).

The second input is GA parameters as presented in Figure 2 (b). GA parameters consist of permutation possibility, crossover possibility, limitation of population, fitness threshold value, number of solutions, and fitness settings. The possibility of permutation and crossover are decimal numbers (0,...,1). The numbers of populations are the maximum number of individu created in the computation process. Populations number also become a stopping condition of the computation process. Fitness threshold is the minimum fitness value to be categorized as the solution. The number of solutions is the number of lecturing schedules as output computation.

FitnessSettings consist of two configurations of some default constraints. They are an option to activate/inactivate default constraints and penalty value for default constraints violation. Default constraints are defined and hardcode constraints in the system.

```
(a) 1 { "days": ["senin", "selasa", "rabu", "kamis", "jumat", "sabtu"],
2   "times": ["08.00 - 08.50", "09.00 - 09.50", "10.00 - 10.50", "11.00 - 11.50", . . . , "19.00 - 19.50"],
3   "rooms": ["R.1&2", "R.3", "R.4", "R.5", "R.6", "R.7", "R.8", "R.9&10"],
4   "roomProps": {
5     "R.1&2": { "owner": ["ilmu hukum", "ilmu hukum (pagi)", "manajemen", "akuntansi"], "capacity": 150 }
6   },
7   "constraints": { "rooms": {}, "lecturers": {}, "prodis": {}, "fixedCourses": [],
8     "others": [
9       { "cond": "(g.time > 3 && g.time < 6 || (g.time < 4 && (g.time + g.sks-1) > 3))", "penalty": 1 },
10      { "cond": "(g.day==4 && g.time >2 && g.time<6) || (g.day==4 && g.time < 3 && (g.time + g.sks - 1) > 2)",
11        "penalty": 1 } ]
12  },
13  "courses": [{ "prodi": "manajemen", "course": "pengantar ekonomi mikro (ma&ak)", "student": 30,
14    "lecturer": "unna ria safitri", "sks": 3, "smt": 1, "required": true }, . . . ]
15 }
```

```
(b) 1 { "nPopulations": 10000, "npops": 10, "nSelection": 3, "pCrossover": 0.85,
2   "pMutation": 0.14, "fitnessThreshold": 0.8, "nSolution": 1,
3   "stoppingCondition": { "nFitnessNoChange": 1000, "fitnessMax": 1 },
4   "fitnessSettings": {
5     "sameLecturerSameTime": { "enable": true, "penalty": 1 },
6     "sameProdiSameSemesterSameTime": { "enable": true, "penalty": 1 },
7     "sameRoomSameTime": { "enable": true, "penalty": 1 },
8     "timeOver": { "enable": true, "penalty": 1 },
9     "roomOverCapacity": { "enable": true, "penalty": 1 },
10    "roomUsedByOthers": { "enable": false, "penalty": 0.01 },
11    "sameLecturerSameDay": { "enable": false, "penalty": 0.001 },
12    "sameLecturerHasSequence": { "enable": false, "penalty": 0.001 },
13    "sameProdiSameSemesterSameDay": { "enable": false, "penalty": 0.001 },
14    "sameProdiSameSemesterHasSequence": { "enable": false, "penalty": 0.001 }, } }
```

```
(c) 1 { "i": 2632, "fitness": 1, "conflict": [],
2   "schedule": [{ "room": "R.3", "day": "selasa", "time": "10.00 - 12.50", "prodi": "manajemen",
3     "lecturer": "unna ria safitri", "smt": 1, "sks": 3, "course": "pengantar ekonomi mikro (ma&ak)",
4     "student": 30, "required": true
5   }, . . . ] }
```

Figure 2 Sample of input & output (a) The input of raw data of lecturing schedule; (b) The input of GA parameters; (c) The schedule output

The output of OPK-GA are schedule which its fitness value upper than fitness threshold. The data structure of the output is presented at Figure 2 (c). There are 4 kinds of information in the output. They are fitness value of individual, the sequence number of individual, unresolved conflict of courses schedule, and lecturing schedule as the computation result. The type of the output is JSON. JSON is widely supported by many program languages. That's why, the other system application such as academic information system, can use the output library easily.

2.4 Default Constraints and Costum Constraints

OPK-GA has 2 types of constraints, they are default constraints (DC) and custom constraints (CC). The CC is defined by the user in the raw data of lecturing schedule at constraints key as presented in Figure 2 (a). The CC consists of string condition and penalty value. The DC is hardcoded constraints in the system. The DC configurations are placed in the parameter GA settings at fitnessSettings as presented in Figure 2 (b). Both DC and CC, fitness value depend on the necessity of constraint to be obeyed. If it must be obeyed, fitness value must be set 1. If the constraint can be ignored or it should be obeyed, so fitness value can be set between 0 – 1.

The DC consists of 2 categories, i.e main default constraints (MDC) and additional default constraints (ADC). The MDC consists of 3 constraints that must be obeyed. They are (1) sameLectureSameTime or a lecturer cannot teach more than one class at the same time; (2) sameProdiSameSemesterSameTime or a course with the same study program and same

semester cannot be scheduled at the same time; (3) sameRoomSameTime or a classroom cannot be used by more than one group user at the same time. The ADC consists of some constraints i.e. (1) timeOver or lecturing must be held in campus operational time; (2) roomOverCapacity or over capacity of classroom is not allowed; (3) roomUsedByOther or all rooms are not free to be used by every study program; (4) sameLectureSameDay or lecturer cannot teach more once a day; (5) sameLectureHasSequence or lecturer cannot teach in a sequence time; (6) sameProdiSame-SemesterSameDay or some course of a study program with the same semester cannot be scheduled on the same day; and (7) sameProdiSameSemesterHasSequence or some course of a study program with the same semester cannot be scheduled in a sequence time;

The MDC and ADC configurations in the GA parameters appropriate to cases, characteristics and constraints in the lecturing Schedule of UBY are represented in Table 2. Appropriate default constraints are activated and its fitness values set appropriate its necessity.

Table 2 Default constraint configuration for UBY lecturing schedule cases

No	UBY Constraints		Key of Default Constraint in GA Parameters	Configurations
	Key	Notes		
1	(C-1)	No crash among lecturers, students and rooms schedule.	sameLecturerSameTime, sameProdiSameSemesterSameTime, sameRoomSameTime, timeOver	Enable = true, penalty = 1
2	(C-2)	12.00 – 13.00 is a break/free time. except on Friday, 11.00 – 13.00	(Defined in custom constraints)	
3	(C-3)	Lecturing begins at 14.00 except for 3rd semester of science law study program with morning class.	(Defined in custom constraints)	
4	(C-4)	Science law study program with morning class must finish lecturing before 12.00 each day.	(Defined in custom constraints)	
5	(C-5)	Every classroom can be used by another study programs,	roomUsedByOthers	Enable =false, penalty = 0
6	(C-6)	Lecturing participants must not over classroom capacity.	roomOverCapacity	Enable = true penalty = 1
7	(C-7)	The activity load per day of students and lecturers are flexible (it can be ignored).	sameLecturerSameDay, sameLecturerHasSequence, sameProdiSameSemesterSameDay, sameProdiSameSemesterHasSequence	Enable =false penalty = 0

Constraints of (C-2), (C-3), dan (C-4) in Table 2 is not accommodated by the library or it is categorized as custom constraints. They must be defined in the raw data of lecturing schedule at constraint part. Figure 3 represents how are they defined.

```

102 | "constraints": {
103 |   "rooms": {},
104 |   "lecturers": {},
105 |   "prodis": { "ilmu hukum (pagi)": { "cond": "g.time > 3", "penalty": 0.01 } }, (C-4)
106 |   "fixedCourses": [],
107 |   "others": [{ "cond": "(g.prodi !== \"ilmu hukum (pagi)\" && g.time < 6)", "penalty": 0.01 }, { (C-3)
108 |     "cond": "(g.time > 3 && g.time < 6 || (g.time < 4 && (g.time + g.sks-1) > 3))", "penalty": 1}, {
109 |     "cond": "(g.day==4 && g.time >2 && g.time<6) || (g.day==4 && g.time < 3 && (g.time + g.sks - 1) > 2)",
110 |     "penalty": 1
111 |   }
112 | },
113 | "courses": []
114 | }
115 |

```

Figure 3 Custom constraint of UBY lecturing schedule cases.

2.5 Computation Design

Computation step of OPK-GA library is presented in Figure 4. Data and Parameter validation begin the computation. That validation process is done by DataValidator Component and ParameterValidator Component. If data and parameter valid, computation will continue to initial population generation by ChromosomeBuilder component.

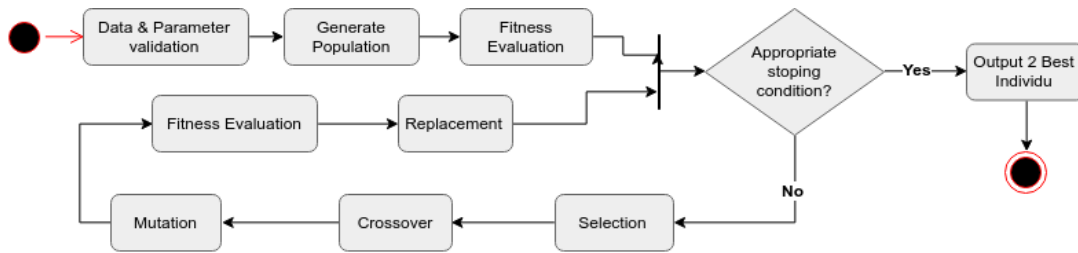


Figure 4 Computation Flow chart of OPK-GA

Next, generated initial population by ChromosomeBuilder will be evaluated by FitnessEvaluator Component. If none individual whose its fitness value upper than fitnessThreshold than process will continue GA Operation by GA-Operator. GA-Operator does a looping processes e.i. selection, crossover, mutation, and fitness value evaluation, and replacement until one of some stopping conditions be triggered. Stopping conditions are (1) minimal there is an individual which its fitness value \geq FitnessMax; (2) Total of generated individual \geq nPopulations (max generated population); (3) there is no changing for the best fitness in a sequence of nFitnessNoChange GA-Operator looping process. The configurations of fitnessMax, nPopulations, and nFitnessNoChange are in GA parameters, as presented in Figure 2 (b).

2.5.1 Gene and chromosome

A chromosome is a sequence of genes [14]. There is 4 information in a gene e.i. a course and a lecturer plotting, time, day and rooms. A course and a lecturer plotting must exist and no changing in every computation. That's why the sequence of courses and lectures plotting is adopted as a sequence of gene position (locus) in a chromosome. Rooms, time and days are chosen as Variance data of gene value (allele).

For example, as shown in Equation 1 the courses and lecturer plotting is symbolized as g as much as h . Rooms data is symbolized as r and as much as i (equation 2). Days data is symbolized as d and as much as j (equation 3). Time data is symbolized as t and it is as much as k (equation 4).

$$g = [0, \dots, g_{h-1}] \tag{1}$$

$$r = [0, \dots, r_{i-1}] \tag{2}$$

$$d = [0, \dots, d_{j-1}] \tag{3}$$

$$t = [0, \dots, t_{k-1}] \tag{4}$$

If gene compositions are variations of room, day and time than a chromosome will be built as shown in figure 4. Genes index position (g) is the sequence of courses and lecturers plotting. Index of x,y and z sequentially are random value of room, day and time index.

g_0			g_1			g_{\dots}			g_{h-1}		
r_{x0}	d_{y0}	t_{z0}	r_{x1}	d_{y1}	t_{z1}	r_{\dots}	d_{\dots}	t_{\dots}	$r_{x(h-1)}$	$d_{y(h-1)}$	$t_{z(h-1)}$

Figure 4 Genes structure in a chromosome

2.5.2 Fitness function

Fitness function ($f(x)$) for chromosome fitness calculation is shown in the equation (5). There is two type of function in the fitness function i.e $V(g)$ and $Y(g,g)$. Function $V(g)$ evaluates

fitness value every gene one by one. $Y(g, g)$ evaluates fitness value every gene by comparing gene's compositions with the other genes' compositions. The variable g is a gene, h and i are the number of genes in the chromosome. The variable m is the total of constraints that involves one gene only. The variable n is the total of constraints that involves 2 genes.

$$f(x) = \frac{1}{1 + \sum_{m=0}^{m-1} \sum_{h=0}^{h-1} V_m(g_h) + \sum_{n=0}^{n-1} \sum_{h=0}^{h-1} \sum_{i=0}^{i-1} Y_n(g_h, g_i)} \quad (5)$$

2.5.3 Selection by a tournament

Selection is a random process to select a parent's chromosome from a population based on fitness value every chromosome [14]. Selection process by a tournament is more efficient than other selection methods and leads to optimal solutions [14]. Before the selection process by tournament begins, it must define the number of participants (N). Then, tournament selection randomly selects N candidate from the population. The winner of the tournament is the chromosome with the highest fitness value. This selection process is repeated until 2 chromosomes selected as a pair of the parent.

2.5.4 Crossover by uniform

Crossover is parents' chromosomes combining process to get new individual. Used method is Uniform Crossover. This method defines biner (0 and 1) mask of locus randomly. As shown in Figure 5 (a), when the mask of locus has value 1, the first spring has gene compositions as same as the first parent, the second spring has gene compositions as same as the second parent. If the mask of locus has value 0, the first spring has the second parent gene compositions and the second spring has the first gene compositions.

Parent 1	1 0 1 1 0 0 1 1
Parent 2	0 0 0 1 1 0 1 0
Mask	1 1 0 1 0 1 1 0
Child 1	1 0 0 1 1 0 1 0
Child 2	0 0 1 1 0 0 1 1

a

Parent	1 0 1 1 0 1 0 1
Child	1 1 1 1 0 0 0 1

b

Figure 5 (a) Crossover by uniform; and (b) Mutation by Interchanging [14]

2.5.5 Interchange Mutation vs Violated Directed Mutation

Mutation is a gene modification process to get a new individual. Two mutation methods try to compare here. They are Interchange Mutation and Violated Directed Mutation (VDM). Interchange Mutation, as shown in Figure 5 (b), switch genes value between two genes with different locus. Different locus/genes position in the chromosome is randomly selected. VDM modifies gene composition (allele) for every violates gene. At first, all violate genes are marked and their allele is reconstructed again [15]. Reconstruction process by the library is generation new allele randomly.

2.5.6 Replacement by Weak Parent Replacement

Replacement is a process to replace selected individual to the population. Weak parent replacement is one of replacement method. This method competes springs and parents fitness value. Two best fitness values are the winner and re-enter to the population for the next GA process. This method maintains the best individual stay in the population.

3. RESULTS AND DISCUSSION

3.1 Interchanging mutation and VDM

A random population generation will produce in a variant of fitness values. At crossover, parents' genes are exchanged. In the small population, it is possible that selected parents have mostly the same genes structures. This situation is not effective. That's why mutation takes place as the biggest part to generate new individual with better fitness value.

Course credits of UBY at first semester of 2017/2018 after filter applying totally has 438 SKS (Tabel 1). Total lecturing slot is 576 (break time is included). It means there are 138 LS free. If only active MDC, the library should produce minimal an optimal solution (appropriate with MDC).

Two mutation method (interchanging and VDM) are tested to get which the best mutation method is. The library configurations for the test are only MDC activated, mutation possibility 0.14, and crossover possibility 0.85. Every mutation method is tested 10 times. The test result is shown in Table 3. As shown in Table 3, Interchanging Mutation never produces an optimum solution. Its fitness value average only 0.02039. But VDM has possibility 1 to produce an optimal solution and its fitness value average is 1. This test results show that VDM is better than interchanging.

Table 3 The test results of Interchanging Mutation vs VDM

Method	Time avg(s)	Cycle avg	Max fitness	Min fitness	Fitness avg	P Output
Interchange	569.529	3365.5	0.02326	0.01667	0.02039	0
VDM	160.629	527.2	1	1	1	1

Notes: cycle: the amount of GA looping process; P output: possibility to produces output

Interchanging mutation exchanges the values (allele) of two genes which its place was randomly selected. It must be remembered that an allele is an array. It consists of a room, day, and time (As shown in Figure 4). The exchange process switches all alleles of one selected gene to another selected gene. The exchange is not array value modifications. It means, there is a high possibility that the violates genes still violates the constraint even if it was exchanged by another gene. Table 3 approves that statement, Interchanging Mutation never produces an optimal solution even if only MDC activated.

Different from interchanging mutation, VDM creates a new variant of the allele of violates genes. The new variant of allele has a possibility to obey the constraint. In this method, non-violates genes are maintained their allele structure. Table 3 approves that VDM more appropriate for scheduling computation when the allele is an array and its variant is randomly selected.

3.2 Lecturing Schedule computation of UBY

Mostly, lecturing in UBY begins at afternoon except for study program of science law at 3rd semester with morning class. After compared, lecturing slot for afternoon class has 228 LS. The credits total for afternoon class is 455 SKS (Table 1). it needs about 167 LS more. This lack makes computation never produces an optimal solution. The results in Table 4 number 1 confirms it. The computation never produces an optimal solution both all constraint activated and the only constraint activated.

The lack of lecturing slot can be resolved by adding some new classroom and adding time to increase lecturing slot in a day. Time for lecturing is from 08.00 to 20.00, it is a long day. So, adding classroom more prefer as priority recommendation to adds lecturing slot than adding new time slot for lecturing. For example, if it is chosen to add some new classroom

which its capacity is 50 person then a new classroom with 6 LS per day in 6 days will give 36 of new LS. If the lack classroom is 161 LS, than it is needed minimal 5 new classrooms (5 room x 6 days x 6 slots = 180 LS). After that, there are 13 LS remain.

The test result which is shown in Table 4, line 6, column "All Constraint" shows that adding 5 new rooms is not enough to produce a solution with fitness value more than 0.8. Although adding 25 new rooms it is not enough to produce an optimal schedule also.

Table 4 The computation result of lecturing schedule with room adding and constraint modifications

No	Room adding	Room Totals	All Constraint			No (C-3)			No (C-3) and (C-6)		
			Time (s)	Cycle	Best-Fitness	Time (s)	Cycle	Best-Fitness	Time (s)	Cycle	Best-Fitness
1	0	8	809.36	1016	0.00582	586.71	1655	0.00769	556.71	1711	0.01053
2	1	9	668.17	794	0.00836	900.49	2570	0.03125	727.39	2254	0.04545
3	2	10	977.66	1186	0.00808	717.36	2034	0.01754	1224.36	3768	0.11111
4	3	11	567.88	688	0.00765	750.15	2134	0.01493	292.44	900	1
5	4	12	598.60	737	0.00716	1014.00	2907	0.02439	707.92	2185	0.03333
6	5	13	487.48	656	0.00743	615.56	1772	0.01515	800.59	2482	0.1
7	6	14	1803.83	2258	0.00994	843.46	2411	0.01639	360.17	1086	(b) 1
8	7	15	1356.71	1402	(a) 0.00904	765.36	2195	0.025	761.24	2313	1
9	12	20	1290.44	1577	0.00956	1057.30	3017	0.02273	150.44	452	1
10	17	25	1449.71	1871	0.01155	1137.38	3356	1	80.46	235	1

Notes: Best-Fitness: The best fitness value after computation; cycle: the numbers of GA looping process
 Constraint (C-3) : Lecturing begins at 14.00 except law science study program at 3rd semester with morning class.
 Constraint (C-6): Lecturing participants must not over classroom capacity.

The next step is result test to know which constraints are mostly violated. The most violated constraints is not included in the next computation to find the possibility of producing optimal solutions. The chosen data for result test is the test results in Table 4, line 8 column "All Constraint". It was 7 new rooms added so there are 15 classrooms. This adding makes new 252 LS (7 rooms x 6 days x 6 hours). If compared with the lack of rooms needs there are 91 LS remains. This remain LS was estimated enough for computation to produce an optimal solution.

The result of result test is shown in Figure 6 (a). It shows that (C-3), (C-2) and (C6) are mostly violated constraints. Constraint (C-3) is lecturing begin at 14.00 except for 3rd semester of science law study program with morning class. Constraint (C-2) is break / free time at 12.00 – 13.00 except on Friday, 11.00 – 13.00. Constraint (C-6) is lecturing participants must not over classroom capacity.

The next test was chosen to not activate (C-3) and (C-6). Constraint (C-3) is the most violated constraint so it must be inactivated. Constraint (C-6) takes effect on some class only. The constraint (C-2) is still activated because it is needed by all study program. This test results:

1. Only constraint (C-3) is not activated. The computation reaches fitness value = 1 when 17 new rooms added. Totally, there are 25 rooms used in the computation (Table 4, row 10, column "No (C-3)").
2. Constraint (C-3) and (C-6) is not activated. The computation reaches fitness value = 1 when minimal 3 new rooms added. There is 11 classrooms use in computations. But the result is not stable because the next test at row 5 & 6 column "No (C-3) and (C-6)" do not produce fitness value = 1. After test number 7 and later the computation results show that computation produces a stable fitness value. This fact confirms that lecturing

schedule of UBY needs minimal 6 new rooms for a stable result and total there are 14 classrooms.

The test result at row 7 column "No (C-3) and (C-6)" is selected for result test. The result of the test is shown in Figure 6 (b). This result shows that there is none violation to MDC and there is none course scheduled at break time. This result is appropriate with condition no (C-3) and (C-6) that there are 69 courses are scheduled in the morning class lecturing slot and 18 course has participants more than room capacity.

```

a
=====
viddia@viddia01 /mnt/dsk02/Researchs/optimasi penjadwalan/programs/
opkga-uby/apps $ node UjiHasil.js
=====
Uji hasil komputasi OPK-GA Library
File : ../schedules//uji5-R17ujihasil/schedule_0.json
Fitness : 0.00904
=====
Hasil:
{ lecturerConflict: 6,
  roomConflict: 24,
  studentConflict: 19,
  restTimeConflict: 31,
  roomOverCapacity: 15,
  timeOverConflict: 0,
  kelasPagiSoreConflict: 63 }
=====

b
=====
viddia@viddia01 ~/Documents/Researchs/optimasi penjadwalan/programs/o
pkga-uby/apps $ node UjiHasil.js
=====
Uji hasil komputasi OPK-GA Library
File : ../schedules/schedules_171213_overcap_false/uji5-R16/schedul
e_0.json
Fitness : 1
=====
Hasil:
{ lecturerConflict: 0,
  roomConflict: 0,
  studentConflict: 0,
  restTimeConflict: 0,
  roomOverCapacity: 18,
  timeOverConflict: 0,
  kelasPagiSoreConflict: 69 }
=====

```

Figure 6 The result of result OPK- GA test; (a) Result test with 7 new rooms adding and activated all constraints; (b) Result test with 3 new rooms adding and inactive constraint (C-3) & (C-6)

3.3 The efficient and effectivity of computation

Computation results in table 4 confirm that the OPK-GA library has good efficiency and effectiveness. Lecturing computation process is fast. The maximum computation with fitness value =1 time takes place in 761.24s or 12 minutes 41 second (Table 4 row 8 column no (C-3) and (C-6)). Effectivity of the library is confirmed, it can produce a solution with fitness value =1 ((C-3) and (C-6) inactive and 6 new rooms added)

3.4 The advantages and disadvantages of the library

By creating raw data of lecturing schedule and GA parameters as shown in Figure 2 (a) and Figure 2 (b) makes library has some advantages e.i. (1) GA Parameters are easier to be configurated as schedule constraints; (2) The constraints that are not provided by the library, can be defined in custom constraints in the raw data of lecturing schedule. By this method, it is possible to add new conditions without any changes in the core of the library. (3) The data type of input and output in JSON makes easier to the developer to give input and process the output of the library.

Some library's lacks are (1) it can not recognize the most violated constraint by itself. When it can recognize by itself, the most violated constraint can be inactivated or reducing its penalty value in the next computation to get an optimal solution. (2) The custom constraints defined by the user have a high possibility to be an error. The user should clearly understand to the whole of schedule constraints and the way how to write it in custom constraints.

4. CONCLUSIONS

The custom constraint in the raw data of lecturing schedule has a possibility to define lecturing schedule constraints that are not provided by the library without any changes in the core of the program. Defining conditions in custom constraints need understanding to the whole schedule's constraints and the way to write custom constraint's condition. The use of VDM in the library is more promising an optimal computation than the use of Interchanging Mutation. The library has good effectivity and efficiency, it can produce an optimal solution with fitness value = 1 in 12 Minutes 41 seconds (in UBY cases the (C-3) and (C-6) are inactivated and 6 new rooms added).

REFERENCES

- [1] Forlap Dikti, “Profil Perguruan Tinggi (Universitas Boyolali),” *Pangkalan Data Pendidikan Tinggi, Kementerian Riset, Teknologi dan Pendidikan Tinggi*, 2017. [Online]. Available: <https://forlap.ristekdikti.go.id/perguruantinggi/detail/NzgxNkEzMzItODhGQi00NDJDLU11OUQtMUU5NDQxMTUyOERG>. [Accessed: 04-Oct-2017].
- [2] A. Darmawan and R. M. Hasibuan, “Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Dengan Mempertimbangkan Team-teaching,” *Simp. Nas. RAPI XIII*, 2014.
- [3] R. Erama and R. Wardoyo, “Modifikasi Algoritma Genetika untuk Penyelesaian Permasalahan Penjadwalan Pelajaran Sekolah,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 8, no. 2, pp. 111–120, Jul. 2014.
- [4] N. L. G. P. Suwirmayanti, I. M. Sudarsana, and S. Darmayasa, “Penerapan algoritma Genetika untuk Masalah Penjadwalan,” *J. Appl. Intell. Syst.*, vol. 1, no. 3, pp. 220–233, 2016.
- [5] C. Setya Pratama, F. Nhita, and A. Aditsania, “Optimasi Penjadwalan Perkuliahan dengan Menggunakan Pendekatan Metode Hybrid Algoritma Genetika Adaptif dan Algoritma Koloni Lebah Buatan (Studi Kasus: Penjadwalan perkuliahan pada Fakultas Teknik Universitas Telkom),” *Indosc 2016*, no. August, pp. 245–256, 2016.
- [6] S. Parera, H. T. Sukmana, and L. K. Wardhani, “Application of Genetic Algorithm for Class Scheduling (Case study: Faculty of science and technology UIN Jakarta),” in *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 2016, pp. 3–7.
- [7] H. P. Hariyadi, T. Widiyaningtyas, M. Z. Arifin, and S. Sendari, “Implementation of Genetic Algorithm to Academic Scheduling System,” in *2016 IEEE Region 10 Conference (TENCON)*, pp. 2013–2016, 2016.
- [8] K. Sylejmani, A. Halili, and A. Rexhepi, “Balancing Academic Curricula by Using a Mutation-only Genetic Algorithm,” *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc.*, pp. 1189–1194, 2017.
- [9] P. M. Chauhan, K. B. Parmar, and M. B. Mendapara, “Solving Time Table Scheduling Problem by Novel Chromosome Representation Using Genetic Algorithm,” *2015 Int. Conf. Circuit, Power Comput. Technol. [ICCPCT]*, pp. 1–6, 2015.
- [10] R. E. Febrita and W. F. Mahmudy, “Modified Genetic Algorithm for High School Timetable Scheduling with Fuzzy Time Window,” *Proc. - 2017 Int. Conf. Sustain. Inf. Eng. Technol. SIET 2017*, vol. 2018–Janua, pp. 88–92, 2018.
- [11] M. Assi, B. Halawi, and R. A. Haraty, “Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem,” *Procedia Comput. Sci.*, vol. 126, pp. 899–906, 2018.

- [12] S. S. A. Alves, S. A. F. Oliveira, and A. R. R. Neto, "A Novel Educational Timetabling Solution Through Recursive Genetic Algorithms," in *2015 Latin America Congress on Computational Intelligence (LA-CCI)*, pp. 1–6, 2015.
- [13] S. Yang and S. N. Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 41, no. 1, pp. 93–106, Jan. 2011.
- [14] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Berlin Heidelberg: Springer, 2008.
- [15] Suyanto, *An Informed Genetic Algorithm for University Course and Student Timetabling Problems*. Berlin, Heidelberg: Springer, 2010.