IMPLEMENTATION OF GENETIC ALGORITHM FOR STUDENT PLACEMENT PROCESS OF COMMUNITY DEVELOPMENT PROGRAM IN UNIVERSITAS GADJAH MADA

Nanang Arfandi and Faizah

Laboratory of Artificial Intelligence
Dept. of Computer Science and Electronics Universitas Gadjah Mada Indonesia

Email: nanang_arfandi@mail.ugm.ac.id

Abstract

Community Development Program (CDP) is a required subject in Universitas Gadjah Mada for undergraduate students to earn a bachelor's degree. One of the big problem in CDP implementation is student placement process into available thematic units. Each unit as much as possible filled by students which have balanced composition based on specified criteria. Balances in gender ratio and clusters composition are needed to have an objective assigned team students in a unit. In this paper, a genetic algorithm is proposed to solve this placement problem. The results of this paper is a simulation in CDP student placement process which using genetic algorithm to produce recommendation units which have good composition.

Keywords: *Community Development Program, criteria, genetic algorithm.*

Abstrak

Program Pengabdian kepada Masyarakat adalah mata kuliah wajib yang diperlukan mahasiswa Universitas Gadjah mada untuk mendapatkan gelar sarjana. Salah satu masalah dalam implementasi program ini adalah proses penempatan mahasiswa menjadi unit-unit tematik yang tersedia. Setiap unit, sebanyak mungkin diisi oleh mahasiswa yang memiliki komposisi yang seimbang berdasarkan kriteria tertentu. Keseimbangan dalam rasio jenis kelamin dan komposisi kelompok diperlukan untuk memastikan obyektivitas penugasan tim dalam unit. Dalam paper ini, algoritma genetika diusulkan untuk memecahkan masalah penempatan. Hasil dari penelitian ini adalah simulasi dalam proses penempatan mahasiswa yang menggunakan algoritma genetika untuk mengahasilkan rekomendasi unit yang memiliki komposisi yang baik.

Kata kunci: *Program Pengabdian Masyarakat, kriteria, algoritma genetika*

1. Introduction

Every undergraduate students in Universitas Gadjah Mada have to participate Community Development Participation (CDP) Program as a required subject. CDP is held on three periods each year. Every period consists about 3000-6000 students who ready to serve and be placed on units. Unit is consists around 20-30 students who has registered as CDP participant. All this time, the student placement process was held manually by putting them into available units for every students. As a result, so many resources such as time and university officers was wasted. Each unit have to reach a good composition. A good composition mean that unit consist students that balance in some criteria such as gender ratio and clusters composition.

Universitas Gadah Mada consists of eighteen faculties. Faculty is a part of cluster. Cluster is group of faculties that have their own ability and competence. There are 4 clusters according to their divisions :

1. *Sainstek*: includes all students from faculty of Biologi, Geography, Mathematics and Science, and Engineering.
2. *Health*: includes all students from faculty of Pharmacy, Medical, and Dentistry.
3. *Sosiohumaniora*: includes all students from faculty of Economics and Business, Philosophy, Law, Humanities, Social and Political Sciences, Psychology.

4. *Agro*: includes all students from faculty of Forestry, Agricultural, Farms, Agricultural Technology, and Veterinary Medicine.

In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic (also sometimes called a metaheuristic) is routinely used to generate useful solutions to optimization and search problems.[1] Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

Therefore, the goals of this paper is to develop a new approach to solve the CDP student placement process by formulating population composition as multiobjective combinational optimization equations and utilize the method of multi objective genetic algorithm. There are few research studies on *Multi Objective Genetic Algorithm*. A detailed review of some solutions using MOGA can be found in [2-6].

2. Mathematical Models

This section will describes the mathematical models and multi objective functions that use as fitness evaluator for each unit in population. The symbols used to model the problem of CDP student placement process are listed as follows.

$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \alpha_4, \beta_4$: a constant value

$f\left(gender\_ratio\right)$: ratio of male and female students

$f\left(Sainstek_{amount}\right)$: allowed amount of students from *Sainstek* cluster in unit

$f\left(Health_{amount}\right)$: allowed amount of students from health cluster in unit

$f\left(Sosiohumaniora_{amount}\right)$: allowed amount of students from *Sosiohumaniora* cluster in unit

$f\left(Agro_{amount}\right)$: allowed amount of students from *Agro* cluster in unit

$\sum male$: total sum of male students

$\sum female$: total sum of female students

$\sum sainstek$: total sum of students from *Sainstek* cluster

$\sum health$: total sum of students from health cluster

$\sum sosiohumaniora$: total sum of students from *Sosiohumaniora* cluster

$\sum agro$: total sum of students from *Agro* cluster

$\sum students$: total sum of students

N : maksimum number of students in unit

2.1. Interval value gender ratio

Each unit has around 20-30 students as team members. First criterion is the balance in gender ratio based on population ratio. The function is :

$$f\left(gender\_ratio\right) = \frac{\sum male}{\sum female} \tag{1a}$$

Then the function (1a) is transformed into interval value gender ratio for each unit.

$$\alpha_0 \le f\left(gender\_ratio\right) \le \beta_0 \tag{1b}$$

2.2. Interval value for amount of students cluster 1

Second criterion is the allowed amount for students from cluster 1(*Sainstek cluster*) in each unit. The function is :

$$f\left(Sainstek_{amount}\right) = \frac{\sum sainstek}{\sum students} \times N \tag{2a}$$

Then it's transformed as function of interval value for allowed amount of *Sainstek cluster* in each unit.

$$\alpha_1 \le f\left(Sainstek_{amount}\right) \le \beta_1 \tag{2b}$$

2.3. Interval value for amount of students cluster 2

Third criterion is the allowed amount for students from cluster 2(*Health cluster*) in each unit. The function is :

$$f\left(Health_{amount}\right) = \frac{\sum health}{\sum students} \times N \tag{3a}$$

Then it's transformed as function of interval value for allowed amount of *health cluster* in each unit.

$$\alpha_2 \le f\left(Health_{amount}\right) \le \beta_2 \tag{3b}$$

2.4. Interval value for amount of students cluster 3

Fourth criterion is the allowed amount for students from cluster 3(*Sosiohumaniora cluster*) in each unit. The function is :

$$f\left(Sosiohumaniora_{amount}\right) = \frac{\sum sosiohumaniora}{\sum students} \times N \tag{3a}$$

Then it's transformed as function of interval value for allowed amount of *sosiohumaniora cluster* in each unit.

$$\alpha_3 \le f\left(Sosiohumaniora_{amount}\right) \le \beta_3 \qquad (3b)$$

2.5. Interval value for amount of students cluster 4

Fifth criterion is the allowed amount for students from cluster 4(*Agro cluster*) in each unit. The function is :

$$f\left(Agro_{amount}\right) = \frac{\sum agro}{\sum students} \times N \qquad (4a)$$

Then it's transformed as function of interval value for allowed amount of *agrocluster* in each unit.

$$\alpha_4 \le f\left(Agro_{amount}\right) \le \beta_4 \qquad (4b)$$

2.6. Multi objective function for each unit

All criteria above must be met by a unit to qualify as a survivor in population. The multi objective function is show on equation 5.

$$J = \begin{bmatrix} f\left(gender\_ratio\right) \\ f\left(Sainstek_{amount}\right) \\ f\left(Health_{amount}\right) \\ f\left(Sosiohumaniora_{amount}\right) \\ f\left(Agro_{amount}\right) \end{bmatrix} \equiv \begin{bmatrix} \alpha_0 \le f\left(gender\_ratio\right) \le \beta_0 \\ \alpha_1 \le f\left(Sainstek_{amount}\right) \le \beta_1 \\ \alpha_2 \le f\left(Health_{amount}\right) \le \beta_2 \\ \alpha_3 \le f\left(Sosiohumaniora_{amount}\right) \le \beta_3 \\ \alpha_4 \le f\left(Agro_{amount}\right) \le \beta_4 \end{bmatrix} \qquad (5)$$

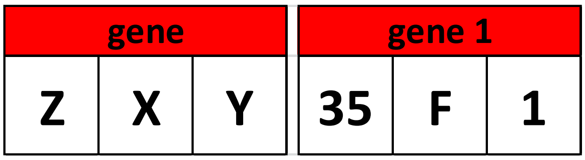
3. The Proposed Method

An interval range value multi objective genetic algorithm was used in this research to solve the CDP student placement process. The process in chromosome encoding, crossover, mutation operators are described as follows.

To solve the units difference on fixed students, first all units have to be ordered based on amount of fixed students. This order is used as *rank selection* for choosing parents in each generation later.

3.1. Chromosome Encoding

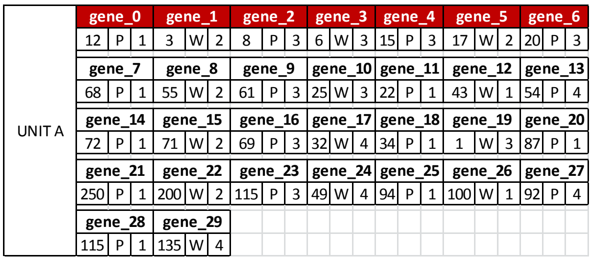
This paper proposes a way to represent student as a single gene. A gene have three sub-genes. Three sub-genes represent variables for a student; sub-gene Z is representing student id, sub-gene X is representing sex type of student (male/female), sub-gene Y is representing the cluster where a student from (1/2/3/4). Then represent of some genes which merge into a single individual called unit. In the end, a group of units is called as population.



gene			gene 1		
Z	X	Y	35	F	1

Fig 1. Gene encoding

Figure 1 shows how a gene presented. The right one is an example of gene. It means that gene 1 has attributes id is 35, sex is female, and from cluster 1. ID attribute is unique, so there is impossible an id could appear twice or more in unit or population. As shown in figure 2 below, a unit is consisted of 30 genes. The interval range value multi objective criteria will be use on a unit like it. If a unit fulfills the criteria, then it's called as a survivor.



	gene_0			gene_1			gene_2			gene_3			gene_4			gene_5			gene_6		
UNIT A	12	P	1	3	W	2	8	P	3	6	W	3	15	P	3	17	W	2	20	P	3
	gene_7			gene_8			gene_9			gene_10			gene_11			gene_12			gene_13		
	68	P	1	55	W	2	61	P	3	25	W	3	22	P	1	43	W	1	54	P	4
	gene_14			gene_15			gene_16			gene_17			gene_18			gene_19			gene_20		
	72	P	1	71	W	2	69	P	3	32	W	4	34	P	1	1	W	3	87	P	1
	gene_21			gene_22			gene_23			gene_24			gene_25			gene_26			gene_27		
	250	P	1	200	W	2	115	P	3	49	W	4	94	P	1	100	W	1	92	P	4
	gene_28			gene_29																	
	115	P	1	135	W	4															

Fig 2. Unit encoding

3.2. Generating Population

Generating population in this paper using random permutation method to get random sequence of students id. All non-fixed students id is choosen as an array data. Then the array is randomized and distributed in available units evenly. In generating population, fitness evaluate based on criteria is performed. Figure 3 shows a flowchart how to generate population.

3.3. Crossover

Parent selection method in this paper is *rank selection*. But it's not an ordinary rank selection. Adapted rank selection was used to fulfill the CDP student placement process. Chromosomes are ranked based on quantity of fixed-students in every units in population. All units which meet the criteria will be flagged as 1 and called a survivor. Meanwhile all units which still not meet the criteria will be flagged as 0 and choosen as parents to produce offspring for next generation. Crossover rate=1 is used, which means all remaining units will pair as parents. Two points crossover do in every sequence pairs of parents. Two point are generated by generating two random gene value of unit.. Figure 4 and 5 gives an ilustration how to implement randomized two point crossover.

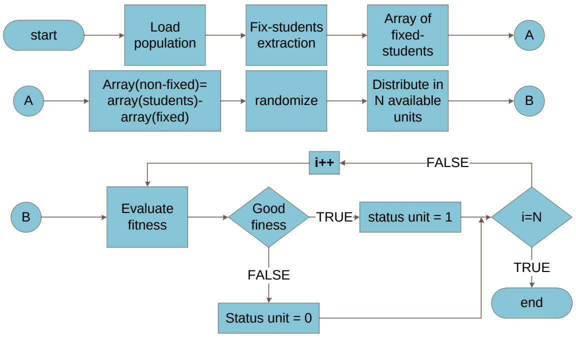


Fig 3. Generating population procedure

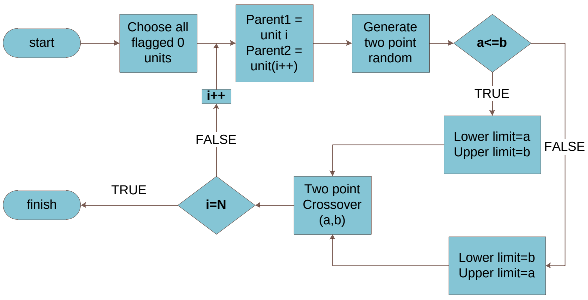
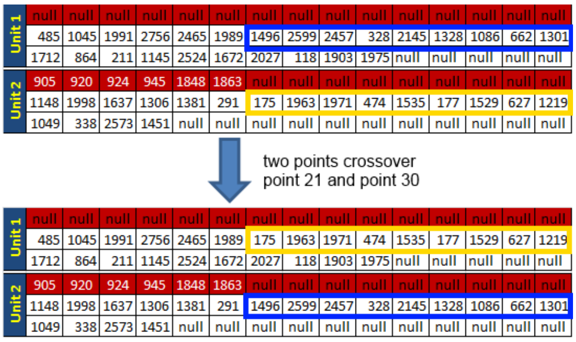


Fig 4. Crossover flowchart



Unit 1	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
	485	1045	1991	2756	2465	1989	1496	2599	2457	328	2145	1328	1086	662	1301
	1712	864	211	1145	2524	1672	2027	118	1903	1975	null	null	null	null	null
Unit 2	905	920	924	945	1848	1863	null	null	null	null	null	null	null	null	null
	1148	1998	1637	1306	1381	291	175	1963	1971	474	1535	177	1529	627	1219
	1049	338	2573	1451	null	null	null	null	null	null	null	null	null	null	null

two points crossover
point 21 and point 30

Unit 1	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
	485	1045	1991	2756	2465	1989	175	1963	1971	474	1535	177	1529	627	1219
	1712	864	211	1145	2524	1672	2027	118	1903	1975	null	null	null	null	null
Unit 2	905	920	924	945	1848	1863	null	null	null	null	null	null	null	null	null
	1148	1998	1637	1306	1381	291	1496	2599	2457	328	2145	1328	1086	662	1301
	1049	338	2573	1451	null	null	null	null	null	null	null	null	null	null	null

Fig 5. Crossover example

Figure 5 is an example a crossover between unit1 and unit2. Point 21 and point 30 are points where crossover happen.

3.4. Mutation

This paper implements *scramble mutation for permutation*. All genes which non-fixed students in each unit area arranged randomly. Mutation happen after crossover finished. Figure 6 gives a flowchart of genetic mutation in this paper and figure 7 is an example of scramble mutation happen.

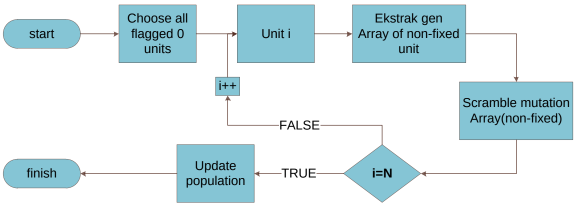
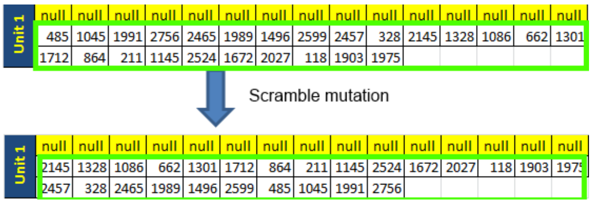

start
Choose all flagged 0 units
Unit i
Ekstrak gen Array of non-fixed unit
Scramble mutation Array(non-fixed)
i=N
TRUE
Update population
finish
FALSE
i++

Fig 6. Mutation schema



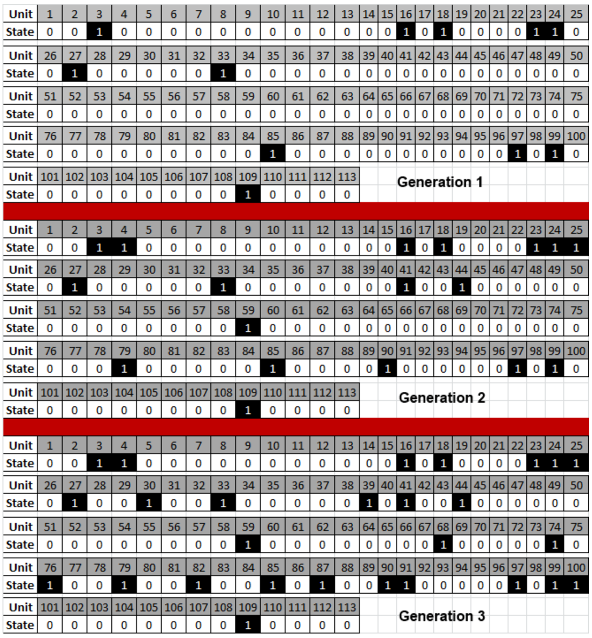
Unit 1	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
	485	1045	1991	2756	2465	1989	1496	2599	2457	328	2145	1328	1086	662	1301
	1712	864	211	1145	2524	1672	2027	118	1903	1975					

Scramble mutation

Unit 1	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
	2145	1328	1086	662	1301	1712	864	211	1145	2524	1672	2027	118	1903	1975
	2457	328	2465	1989	1496	2599	485	1045	1991	2756					

Fig 7. Scramble mutation

After crossover and mutation, generational update occurred in a simple way. Crossover and mutation would produce offsprings which automatically replace all parents in previous generation. After replacing all parents, a new generation has formed. Evaluation on each unit is performed again. Good unit will be flagged as 1, and bad unit will be applied as parent again to produce offspring for next generation. This kind of generational insertion is called *steady-state generation update* model. Figure 8 gives sample about three generation happen.



Unit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
State	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0

Unit	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
State	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Unit	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
State	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Unit	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
State	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Unit	101	102	103	104	105	106	107	108	109	110	111	112	113
State	0	0	0	0	0	0	0	0	1	0	0	0	0

Generation 1

Unit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
State	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1

Unit	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
State	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0

Unit	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
State	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Unit	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
State	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0

Unit	101	102	103	104	105	106	107	108	109	110	111	112	113
State	0	0	0	0	0	0	0	0	1	0	0	0	0

Generation 2

Unit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
State	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1

Unit	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
State	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0

Unit	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
State	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

Unit	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
State	1	0	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1

Unit	101	102	103	104	105	106	107	108	109	110	111	112	113
State	0	0	0	0	0	0	0	0	1	0	0	0	0

Generation 3

Fig 8. Three generational update example

4. Experiments and Result Analysis

In this Section, the experiment is demonstrated. Table 1 gives a statistics fact about composition of the data for this experiment.

TABLE I
STATISTICS FACT

Facts	Informations
Total students	2791 students
Total Unit	113
Fixed students	605 students
Non-fixed students	2186 students
Average of students in unit	24 - 25 students/unit
Male students	1273 students
Female students	1518 students
Ratio male/female	0.838603426
Sainstek students	1071≅ [7~10] Students/Unit
Health cluster students	399≅ [1~4] Students/Unit
Sosiohumaniora students	898≅ [6~9] Students/Unit
Agro students	423≅ [2~5] Students/Unit

The algorithm performance was demonstrated in this experiment. Performance is determined by iterations. How many iterations to produce a population that give an optimum result. Figure 9 shows how genetic algorithm performance in the process of CDP student placement. Every iteration period produce some remaining units that don’t balance yet. More small remain units which unbalance in population mean the algorithms more better.

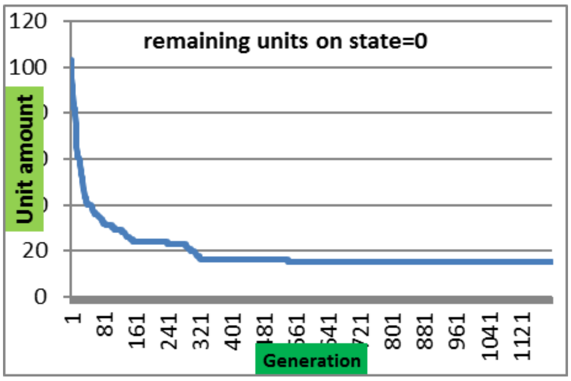


Fig 9. Experiment with mutation rate 1

Experiment with mutation rate 1 needs 1200 iteration for generation update to produce 15 unbalance remaining unit. To prove that mutation rate will affect performance of algorithm. Mutation rate with lower value (0.5) is choosen. The result is experiment with mutation rate 0.5 needs 1600 iteration for generation updateto produce 21 unbalane remaining units. It’s a proove that mutation rate will affect the algorithm performance.

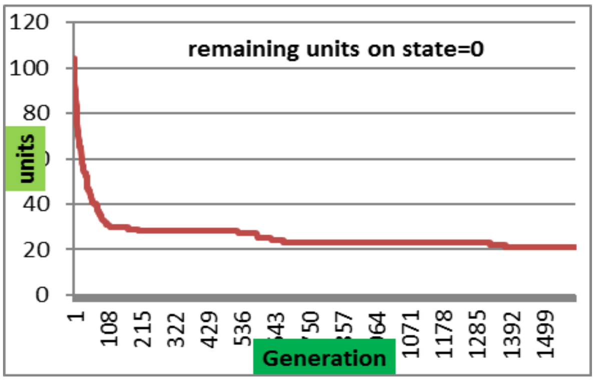


Fig 10. Experiment with mutation rate 0.5

5. Conclusions and Future Work

In the presented paper, a new approach was proposed to solve Community Development Participation student placement process in Universitas Gadjah Mada Indonesia. The approaches was based on Multi Objective Genetic Algorithms schema. In this paper, multi objective criteria is represented as multi objective functions. The multi objective functions formula is transformed into multi objective interval range value functions. This transformation is used for multi criteria evaluation fitness for team members composition in a unit in CDP. This method could be the first approach for this kind of specifique problem. Another problems or cases which have same characteristics could implement this method for solving multi criteria optimization. The main idea of this paper is how we could solve a multi criteria optimization problem using multi objective genetic algorithms. A new proposed method in *MOGA* was given by turning multi objective functions into multi interval range value optimization. Interval range value is allowed values that an objective function could have. As long as chromosomes having all allowed values in multi objective criteria, then it’s a survivor in the population.

Future work could implement some other algorithms method to solve CDP student placement process such as *fuzzy*, some others genetic algorithms (GA’s) method or maybe another brand new initial method. Other methods implementaton for this problem aim to compare the performance between method that is choosen for optimize CDP student placement process.

References

[1] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996. ISBN 9780585030944.

[2] D. Monoyios and K. Vlachos, "Multiobjective Genetic Algorithms for Solving the Impairment-Aware Routing and Wavelength Assignment Problem", *J. OPT. COMMUN. NETW*, VOL. 3, NO. 1, JANUARY 2011, pp. 40-47.

[3] T-Y. Chou, C-N. Lee, T-K. Liu, C-H Chen, and F-S. Chang, "Dynamic Size-based Multiobjective Genetic Algorithm to Solve the Crew Pairing Problem", *SICE Annual Conference*, The Grand Hotel, Taipei, Taiwan, August 2010.

[4] K.Deb, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", unpublished.

[5] K.Deb, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", *IEEE Trans on Evolutionary Computation*, Vol.6, No.2, April 2002, pp. 182-197.

[6] E.I. Ducheyne, R.R. De Wulf, B. De Baets, "A spatial approach to forest-management optimization: linking GIS and multiple objective genetic algorithms", *international Journal of Geographical Information Science*, Vol.20, No.8, September 2006, pp. 917-928.