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.

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. 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.

\[ \sum \text{students} \]: total sum of students  
\( N \): maximum 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:

\[ f(\text{gender\_ratio}) = \frac{\sum \text{male}}{\sum \text{female}} \]  

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

\[ \alpha_0 \leq f(\text{gender\_ratio}) \leq \beta_0 \]  

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:

\[ f(\text{Sainstek\_amount}) = \frac{\sum \text{sainstek}}{\sum \text{students}} \times N \]  

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

\[ \alpha_1 \leq f(\text{Sainstek\_amount}) \leq \beta_1 \]  

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:

\[ f(\text{Health\_amount}) = \frac{\sum \text{health}}{\sum \text{students}} \times N \]  

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

\[ \alpha_2 \leq f(\text{Health\_amount}) \leq \beta_2 \]  

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:

\[ f(\text{Sosiohumaniora\_amount}) = \frac{\sum \text{sosiohumaniora}}{\sum \text{students}} \times N \]
Then it’s transformed as function of interval value for allowed amount of sosiohumaniora cluster in each unit.

\[ \alpha_3 \leq f(Sosiohumaniora_{\text{amount}}) \leq \beta_3 \]  

### 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:

\[ f(Agro_{\text{amount}}) = \sum \frac{agro \times N}{\text{students}} \]  

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

\[ \alpha_4 \leq f(Agro_{\text{amount}}) \leq \beta_4 \]

### 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.

\[
J = \begin{bmatrix}
    f(\text{gender\_ratio}) \\
    f(Sainstek_{\text{amount}}) \\
    f(Health_{\text{amount}}) \\
    f(Sosiohumaniora_{\text{amount}}) \\
    f(Agro_{\text{amount}})
\end{bmatrix}
\]  

\[ \alpha_1 \leq f(\text{gender\_ratio}) \leq \beta_0 \]  
\[ \alpha_1 \leq f(Sainstek_{\text{amount}}) \leq \beta_1 \]  
\[ \alpha_2 \leq f(Health_{\text{amount}}) \leq \beta_2 \]  
\[ \alpha_3 \leq f(Sosiohumaniora_{\text{amount}}) \leq \beta_3 \]  
\[ \alpha_4 \leq f(Agro_{\text{amount}}) \leq \beta_4 \]  

### 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 encoding](image1)

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.

![Unit encoding](image2)

### 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.
Nanang Arfandi, et al., Implementation of Genetic Algorithm

73

...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 illustration how to implement randomized two point crossover.

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.

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.

Fig 3. Generating population procedure

Fig 4. Crossover flowchart

Fig 5. Crossover example

Fig 6. Mutation schema

Fig 7. Scramble mutation

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 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](image)

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 update to produce 21 unbalance remaining units. It’s a prove that mutation rate will affect the algorithm performance.

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 specific 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 implementation 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.