International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Multi-Objective Optimization of Laboratory Technicians Scheduling using Binary Genetic Algorithm

Antoni Wibowo, Filbert Ivander

Abstract*: A laboratory needs at least one technician to maintain the laboratory's activity every day. The technicians should prevent any technical interference in a daily learning activity. The technicians must be placed in a different lab the next day to check the work of the technician previously. This scheduling model has assigned 4 technicians into 3 laboratories in a month. We proposed a mathematical model for multi-objective optimization of laboratory technicians scheduling since it has many objective functions such as avoid collisions, workload balancing of technicians, and works distribution in the laboratories. We presented a Binary Genetic Algorithm to find the best technicians scheduling that can be used to support daily operations. As a result, we noticed that Binary GA could effectively be used in daily operational since the computing time was relatively short in finding the best laboratory technicians scheduling. From ten times of testing, the best solution needs 285.406s to calculate with the minimum function value is 2.

Keywords*: Genetic Algorithm, Laboratory Technicians Scheduling, Binary GA, Multi-objective.

I. INTRODUCTION

Scheduling is the allocation of resources at a specific time that meets the constraints as much as possible to achieve the desired goals [1]. [2][3] provides an overview of various staff and rostering scheduling that have different constraints to solve and satisfy their needs. They also presented the scheduling problem into methods and models. The scheduling process is divided into several modules into one procedure, which consists of determining staff requirements, leave days, changing work shifts, tasks based on skill or levels, and staff assignments. Scheduling development can use only a few modules, depending on the scheduling problem that will be applied. [4] presented the algorithm with a simple module-based heuristic that can solve scheduling in four real-world companies that produce good, fair, and feasible schedules.

Manuscript published on November 30, 2019.
* Correspondence Author
* Correspondence Author
Antoni Wibowo*, Computer Science Department, Bina Nusantara University, BINUS Graduate Program-Master in Computer Science, Jakarta, 10480, Indonesia
Filbert Ivander, Computer Science Department, Bina Nusantara University, BINUS Graduate Program-Master in Computer Science, Jakarta, 10480, Indonesia.

The laboratory is one of the facilities at the educational institution/university/hospital that support their daily activities. Every day, students/patients are scheduled based on shifts in the use of a laboratory. Many problems can occur during the use of a laboratory, such as a computer attacked by a virus which is caused by sending data through a USB flash drive and damage to computer components to cause a total failure of the computer system or medical equipment has a problem in the hospital. Also, we must maintain the cleanliness and comfort of the laboratory as well. Therefore, it is necessary to place a technician in each laboratory.

Every day, a technician should check facilities before and after using the laboratory to ensure the laboratory's equipment is always in good condition. A technician must understand how to maintain a laboratory in the short and long term. However, each technician has a different skill for fixing every computer problem. The allocation technicians in the different laboratories to check the computer problem that had been done by the technician on the previous day. The technician will report problems that occur every month.

In this paper, we proposed a mathematical model for multi-objective optimization of laboratory technicians scheduling since it has many objective functions such as avoid collisions, workload balancing of technicians, and works distribution in the laboratories. to finding the best solution with many objectives is difficult, we transformed the multi-objective optimization problem into a single-objective optimization problem. The best solution in scheduling can generate for each month.

In the real world, finding multi-objective problems is not accessible due to multiple conflicting objectives and the complexity of the problems [5]. In a multi-objective optimization algorithm, the solutions are identified using the Pareto optimal set, but not many multi-objective problems can be identified because the size of the solution is considerable [6].

We consider laboratory technicians scheduling with four technicians and three laboratories as our case study. This scheduling will be arranged periodically every month. Some problems in laboratory technicians scheduling are a few technicians on duty in daily operations except Sundays/national holidays. Every technician is not allowed to work at the same laboratory for the next consecutive day, and the maximum day the work of every technician is 21 days in a month. Also, a technician can request a day off on a specific day and get off on Sundays/national holidays.

Manually arranging a schedule will be difficult, spending much of time and limited in choosing a feasible solution without focusing on optimization. Manually scheduling can also make some violations and unfairly assigned from each technician. Therefore, laboratory technicians scheduling needs to be made automatically and avoid problems in the scheduling. In this paper, we transform the multi-objective optimization problem into a single objective optimization problem to find the best solution of technicians scheduling.

Furthermore, we performed a variant of genetic algorithm (GA) to find the best laboratory technicians scheduling. Chromosomes of GA using Binary number. In other words, our problem is category into a 0-1 multi-objective optimization problem. We used GA as our basis solver since GA is relatively widely used in many domain problems, powerful in solving the optimization problem and easy to implement.

Laboratory technicians scheduling is one of scheduling an educational institution/university other than class timetabling [7], courses timetabling [8][9] and lectures timetabling [10][11]. Besides, scheduling staff has also been applied in various areas such as nurseries [12][13], banks [14], and railways [15]. Various scheduling techniques can be chosen to make effective and efficient scheduling such as genetic algorithms, linear programming [12], integer programming 0-1 [13], heuristics [7][11], and mathematical programming [14].

One of an algorithm that is very powerful in completing optimization is Genetic Algorithm [16]. An automatic timetable using a genetic algorithm can save much time from the administrator in creating and providing optimal solutions [10]. The genetic algorithm has been solving employee scheduling problems with three intervals in a day, for all weekdays at mall type shops, that minimizes errors and obtained hourly/daily coverages [17].

The genetic algorithm has also been applied in the scheduling of resources in well-services companies that solving call problems and people scheduling for working using two methods that are solving before deadline and makespan minimization [17]. [18] introduces the new application in a hostel that allocates bed space available for students that will achieve four-point TREE objectives of transparency, reliability, efficiency, and effectiveness using genetic algorithms. The genetic algorithm has been developed using a C programming language [19], Java programming language [1], PHP [10], and C# programming language [20].

[21] presented a multi-objective optimization model for scheduling construction projects under extreme weather with three major phases that are initialization phase, fitness evaluation phase, and population generation phase. They presented the scheduling that can control the complexity optimization model in are combination crew formation option (daily production and cost rates) into a single variable, and the starting date variable is amount day in a year. To efficiency, multi-objective genetic algorithm for solving the transportation problems are exploring the less-crowded area to obtain more non-dominated solutions using local search technique as a neighborhood that will increase the speed-up search process [22].

II. MATERIAL AND METHOD

A. Mathematical Model

The formulation model is minimizing the working variant of distribution is based on the number of working days and the working variant of distribution each technician in each laboratory every month.

Notation:

I = number of technicians = 4
J = number of laboratories = 3
K = number of days = 31 (based on the number of days in a month)
i = index of technician {1, 2, 3, 4}
j = index of computer laboratory {1, 2, 3}
k = index of working days {1, 2, ..., K}

Decision variables:

$$X_{i,j,k} = \begin{cases} 1, & \text{if technician } i \text{ is assigned at computer lab } j \text{ at day } k \\ 0, & \text{if the opposite} \end{cases}$$

Objective function:

$$\text{Min}\, Z = Z_0 + Z_1 + Z_2 + Z_3 + 100 * Z_4$$

Where:
Z_0 = variant work of distribution based on days
Z_1 = variant work of distribution in laboratory 1
Z_2 = variant work of distribution in laboratory 2
Z_3 = variant work of distribution in laboratory 3
Z_4 = total violations

Constraints:

1) There are only three technicians that work every day except Sunday/national holiday.

a) There are only three technicians that work every day.

$$\sum_{j=1}^{3} X_{1,j,k} + X_{2,j,k} + X_{3,j,k} + X_{4,j,k} = 3$$ for every k = 1, 2, …, 31 (except Sunday/National holiday)

b) All technicians get off on Sunday/National holiday.

$$\sum_{j=1}^{3} X_{1,j,k} + X_{2,j,k} + X_{3,j,k} + X_{4,j,k} = 0$$ for every k = Sunday/national holiday

2) There is only one technician on duty in each laboratory every day except Sunday/National holiday.

a) There is only one technician on duty in each laboratory every day.

$$\sum_{j=1}^{3} X_{1,j,k} + X_{2,j,k} + X_{3,j,k} + X_{4,j,k} = 1$$ for every k = 1, 2, …, 31 (except Sunday/National holiday)

b) All technicians get off on Sunday/National holiday.

$$\sum_{j=1}^{3} X_{1,j,k} + X_{2,j,k} + X_{3,j,k} + X_{4,j,k} = 0$$ for every k = Sunday/National holiday

c) Technician 1 can only work in one laboratory every day.

$$\sum_{k=1}^{31} X_{1,1,k} + X_{1,2,k} + X_{1,3,k} \le 1$$

d) Technician 2 can only work in one laboratory every day.

Retrieval Number: D7204118419/2019©BEIESP
DOI:10.35940/ijrte.D7204.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$\sum_{k=1}^{31} X_{2,1,k} + X_{2,2,k} + X_{2,3,k} \leq 1$$

e) Technician 3 can only work in one laboratory every day.

$$\sum_{k=1}^{31} X_{3,1,k} + X_{3,2,k} + X_{3,3,k} \leq 1$$

f) Technician 4 can only work in one laboratory every day.

$$\sum_{k=1}^{31} X_{4,1,k} + X_{4,2,k} + X_{4,3,k} \leq 1$$

3) Every technician does not work at the same laboratory for the next day consecutively.

a) If technician *i* was assigned at the laboratory 1 on the previous day, then the next day may not work at the laboratory 1 again.

$$\sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,1,k} + X_{i,1,k+1} \leq 1$$

b) If technician *i* was assigned at the laboratory 2 on the previous day, then the next day may not work at the laboratory 2 again.

$$\sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,2,k} + X_{i,2,k+1} \leq 1$$

c) If technician *i* was assigned at the laboratory 3 on the previous day, then the next day may not work at the laboratory 3 again.

$$\sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,3,k} + X_{i,3,k+1} \leq 1$$

4) The maximum day the work of every technician is 21 days in a month.

$$\sum_{i=1}^{4}\sum_{j=1}^{3}\sum_{k=1}^{31} X_{i,j,k} + X_{i,j,k} + \ldots + X_{i,j,k} \leq 21$$

5) Each technician can request a day off on a certain day, the decision variable that related is being 0 (i.e. the technician 1 request the day off on August, 6th 2018, the technician 2 request the day off on August, 9th 2018 and the technician 4 request the day off on August, 13th 2018).

$$X_{1,j,6} = 0, X_{2,j,9} = 0, X_{4,j,13} = 0$$

6) The number of days that the technician works in a month.

$$Y_i = \sum_{i=1}^{4}\sum_{j=1}^{3}\sum_{k=1}^{31} X_{i,j,k}$$

7) The average working day of the technician in a month.

$$MeanY = \frac{Y_1 + Y_2 + Y_3 + Y_4}{4}$$

8) The number of days that the technician *i* work in laboratory 1.

$$W1_i = \sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,1,k}$$

9) The average number of technician *i* work in laboratory 1.

$$MeanW1 = \frac{W1_1 + W1_2 + W1_3 + W1_4}{4}$$

10) The number of days that technician *i* work in laboratory 2.

$$W2_i = \sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,2,k}$$

11) The average number of technician *i* work in laboratory 2.

$$MeanW2 = \frac{W2_1 + W2_2 + W2_3 + W2_4}{4}$$

12) The number of days that the technician i work in laboratory 3.

$$W3_i = \sum_{i=1}^{4}\sum_{k=1}^{31} X_{i,3,k}$$

13) The average number of technician *i* work in laboratory 3.

$$MeanW3 = \frac{W3_1 + W3_2 + W3_3 + W3_4}{4}$$

14) The work variant of distribution based on the number of working days.

$$Z_0 = \sum_{i=1}^{4} \frac{(Y_i - MeanY)^2}{4}$$

15) The work variant of distribution in laboratory 1.

$$Z_1 = \sum_{i=1}^{4} \frac{(W1_i - MeanW1)^2}{4}$$

16) The work variant of distribution in laboratory 2.

$$Z_2 = \sum_{i=1}^{4} \frac{(W2_i - MeanW2)^2}{4}$$

17) The work variant of distribution in laboratory 3.

$$Z_3 = \sum_{i=1}^{4} \frac{(W3_i - MeanW3)^2}{4}$$

18) Violation constraints from number 1-4.

$$Z_4 = \sum_{k=1}^{31(\text{not s\&n})} viocont1A_k + \sum_{k=1}^{s\&n} viocont1B_k + \sum_{j=1}^{4}\sum_{k=1}^{31(\text{not s\&n})} viocont2A_{jk} + \sum_{j=1}^{4}\sum_{k=1}^{s\&n} viocont2B_{jk} + \sum_{k=1}^{31} viocont2C_k + \sum_{k=1}^{31} viocont2D_k + \sum_{k=1}^{31} viocont2E_k + \sum_{k=1}^{31} viocont2F_k + \sum_{i=1}^{4} viocont3A_i + \sum_{i=1}^{4} viocont3B_i + \sum_{i=1}^{4} viocont3C_i + \sum_{i=1}^{4} viocont4_i$$

For every *s* = Sunday and *n* = National holiday

B. Genetic Algorithm

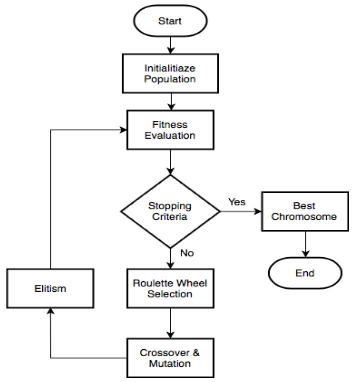

Fig. 1. Genetic Algorithm Flowchart

Figure 1 explains the stage of the Genetic Algorithm until getting a solution in every testing. Following the several stage Genetic Algorithm:

1) Initialize Population

Figure 1 is a Genetic Algorithm flowchart we used in this paper that the beginning starts from the initializing population.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

We representation each chromosome value randomly in Binary 0 or 1 within dimension array (4 technicians x 3 laboratories x 31days = 372 array).

2) Fitness Evaluation

Each chromosome raised in a population will be evaluated to determine its fitness value. Table I shows an example of the day's technicians assigned in a month.

Table - I: An example of the day's technicians Assigned

i	X_i
1	20
2	20
3	19
4	19

In this below is an example process to calculate each variant in equation 14-17. After getting the variant and total violation value, we calculate the objective function.

$$\sum_{i=1}^{4} X_i = 78$$

$$\sum_{i=1}^{4} X_i^2 = 1522$$

$$\left(\sum_{i=1}^{4} X_i\right)^2 = (78)^2 = 6084$$

$$s^2 = \frac{n\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2}{n(n-1)}$$

$$s^2 = \frac{(4.1522) - 6084}{4(4-1)}$$

$$s^2 = 0.333333$$

1. Obj_func (chromosome[1]) = 1.6666666666667 + 9 + 3.6666666666667 + 9.6666666666667 + 100 * 16 = 1624
2. Obj_func (chromosome[2]) = 9.6666666666667 + 3 + 11 + 4.3333333333333 + 100 * 10 = 1028
3. Obj_func (chromosome[3]) = 3 + 6.3333333333333 + 8.3333333333333 + 1.6666666666667 + 100 * 21 = 2119.3333333333
4. Obj_func (chromosome[4]) = 3 + 6.3333333333333 + 9.6666666666667 + 3 + 100 * 19 = 1922
5. Obj_func (chromosome[5]) = 11.666666666667 + 3 + 4.3333333333333 + 6.3333333333333 + 100 * 16 = 1625.3333333333
6. Obj_func (chromosome[6]) = 3.6666666666667 + 7 + 3.6666666666667 + 3 + 100 * 11 = 1117.3333333333
7. Obj_func (chromosome[7]) = 0.33333333333333 + 3 + 3 + 5.6666666666667 + 100 * 19 = 1912
8. Obj_func (chromosome[8]) = 3 + 4.3333333333333 + 9.6666666666667 + 13.666666666667 + 100 * 17 = 1730.6666666667
9. Obj_func (chromosome[9]) = 3 + 9.6666666666667 + 23 + 13.666666666667 + 100 * 10 = 1049.3333333333
10. Obj_func (chromosome[10]) = 3 + 6.3333333333333 + 11 + 7 + 100 * 18 = 1827.3333333333

3) Stopping Criteria

The testing will terminate if the criteria are satisfied and will return the best chromosome.

4) Roulette Wheel Selection

The most general selection in GA is using a roulette wheel [9]. In a roulette wheel, the most fitness value will get a more significant area selected. Figure 2 is the example of roulette wheel selection.

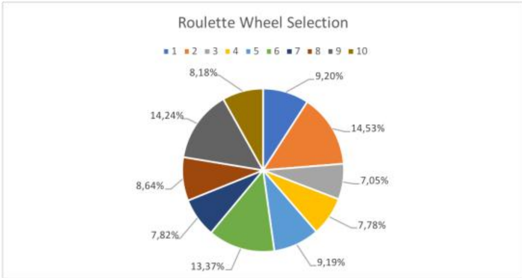

Fig. 2. Roulette Wheel Selection

Here are the several phases in the roulette wheel [8]:

1. Calculate the fitness value in every chromosome.
2. Calculate the fitness value total in a generation.
3. Calculate the selection probability.
4. Calculate the cumulative probability.
5. The selection process generates a new population.

5) Crossover and Mutation

In the crossover phase, we used a single-point crossover and defined the cut-off point from 14th until the end of the days. Figure 3 shows the crossover process of crossing the values to another chromosome that is selected.

The mutation process is randomly selecting only one index and replace the value by randomly. Figure 4 shows the example mutation selection in index 7th values.

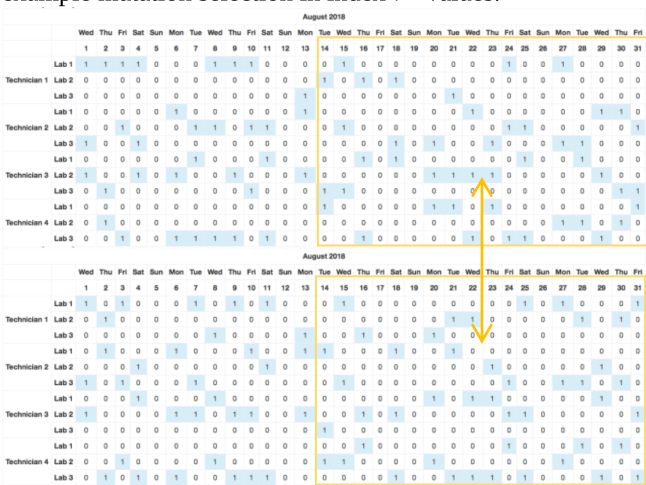

Fig. 3. Crossover

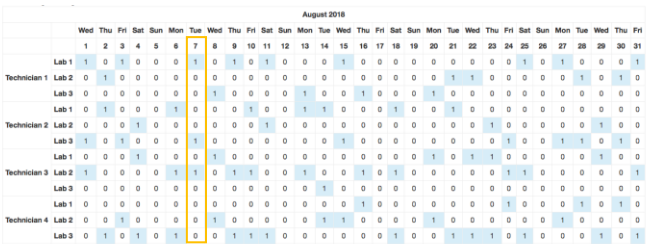

Fig. 4. Mutation

6) Elitism

To keep the best fitness value, we used elitism to store every best fitness value in each iteration.

If the best fitness value in the next generation is less than elitism value, then the elitism value will be replaced by a new one.

7) Best Chromosome

The best chromosome in every test is the optimal solution for scheduling.

III. RESULT AND DISCUSSION

We used Binary GA to solve this scheduling problem. The mathematical model was implemented using the PHP programming language, and experiments were done using a web browser Google Chrome. The experiments used Windows 7, processor Intel(R) Core (TM) i5-4590 CPU @ 3.30 GHz, and memory 8 GB. For the limitation of the scheduling process, we defined chromosomes number is 30 in every population, the maximum generation is 150th. We also set the crossover probability value of 0.75, and the mutation probability value of 0.075.

Table II shows the results experiment of the laboratory technicians scheduling. When the Binary genetic algorithm stops finding the solutions at less than of the 150th iteration, we obtained a minimum value of objective/fitness function and the best iteration/generation. If a violation happened, then the objective value function is higher than 100, the value of the best generation is equal to maximum generation value and we notice the testing value solution is no. It means that the best solution will achieve more than the 150th iteration. But, if all constraints were satisfied, the minimum function value would be less than 100, we found a solution the best generation at the last generation, and we notice the testing value solution is yes. The computation time of our experiments based on either how long the search solution was found or stopped at the 150th iteration.

Table - II: Several Experiment in Laboratory Technicians Scheduling

Experimen t	Probability Crossover	Probability Mutation	Minimum Function	Best Generation	Time (s)	Solution
1	0.75	0.075	101.333	96	278.661	No
2	0.75	0.075	2	146	285.406	Yes
3	0.75	0.075	2.66667	145	290.789	Yes
4	0.75	0.075	204.667	146	282.812	No
5	0.75	0.075	4	127	293.878	Yes
6	0.75	0.075	204	117	285.885	No
7	0.75	0.075	205.333	135	240.869	No
8	0.75	0.075	2.66667	135	284.042	Yes
9	0.75	0.075	202.667	78	293.423	No
10	0.75	0.075	2.66667	99	281.242	Yes

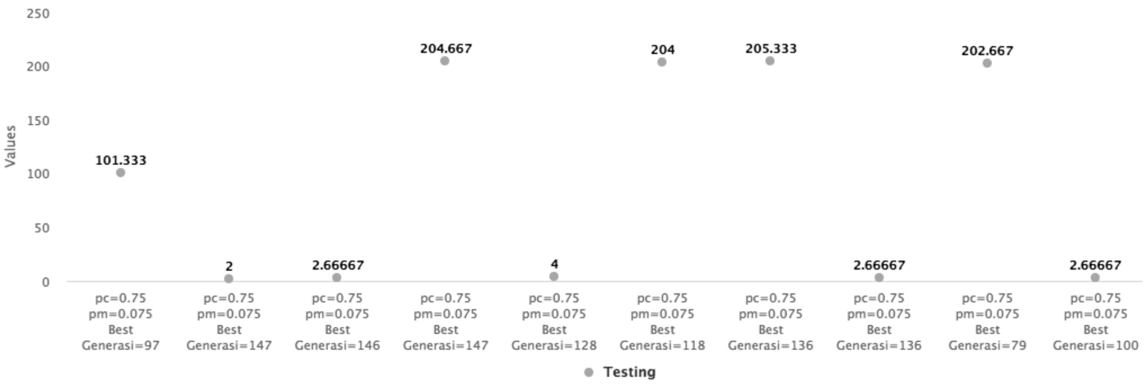

Fig. 5. Minimum Value of Fitness Function

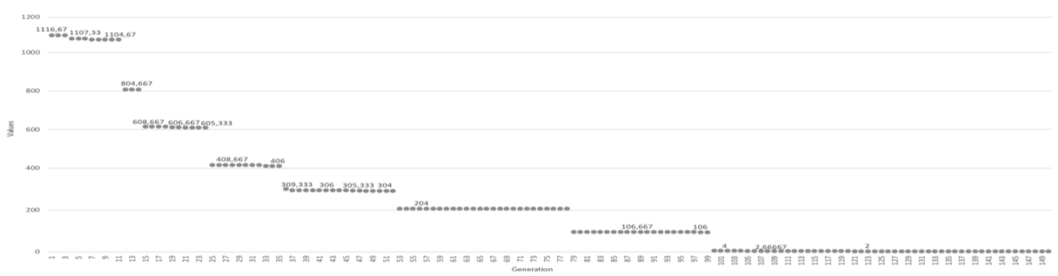

Fig. 6. The Best Values of Fitness Function on The Corresponding Best Generation

Retrieval Number: D7204118419/2019©BEIESP
DOI:10.35940/ijrte.D7204.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

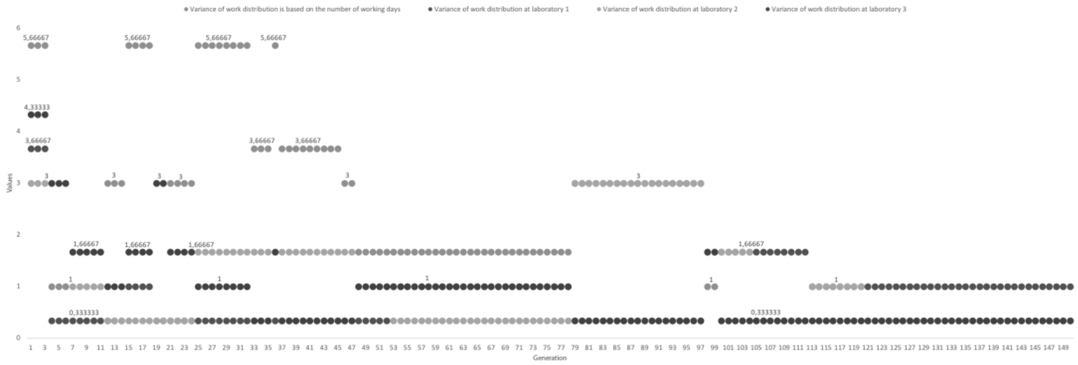

Fig. 7. The Variant Work of Distribution

Table - III: The Best Laboratory Technicians Scheduling Using Binary GA from the 2nd Experiment

		August 2019															
		Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Technician 1	Lab 1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0
	Lab 2	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
	Lab 3	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
Technician 2	Lab 1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
	Lab 2	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0
	Lab 3	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0
Technician 3	Lab 1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0
	Lab 2	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0
	Lab 3	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1
Technician 4	Lab 1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0
	Lab 2	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	Lab 3	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

		Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Technician 1	Lab 1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0
	Lab 2	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1
	Lab 3	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Technician 2	Lab 1	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0
	Lab 2	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
	Lab 3	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0
Technician 3	Lab 1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
	Lab 2	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
	Lab 3	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1
Technician 4	Lab 1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	
	Lab 2	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0
	Lab 3	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0

Based on information in Table II, from 10 times testing, there were 5 times successful testing get resulted in the solution of scheduling and 5 times unsuccessful testing which has several violation constraints. The highest minimum function was 205.333 took computing time 240.869 second that was too short other than the smallest minimum function was 2 which have taken computing time 285.406 second. Figure 5 shows a graph of our experiment that has been carried out. We used the same probability crossover and mutation in every testing, but not every time testing found solution scheduling because GA worked randomly to finding the solution.

Figure 6 shows the highest minimum function value in the first generation worth 1116.67, but after going through the process of improvement in the next generation, the minimum function value was changed to 2 in the 121st generation until the max generation that is 150th generation.

Figure 7 shows that the working variant of distribution based on the number of working days value is 0.3. The work variant of distribution at laboratory 1 is higher than the other which means more technicians are placed in laboratory 1. Table III shows the best solution obtained from generating technician scheduling in August 2018, where all constraints are satisfied.

In the previous paper, GA successfully used to solve scheduling problems in various fields as well as in this laboratory technicians scheduling. The GA operator is beneficial and took a short time in finding good solutions other than people that must find it manually. In future research, the scheduling should consider technician skills, fatigue, and personality [23].

IV. CONCLUSION

We have implemented the Binary Genetic Algorithm for technicians laboratory scheduling using PHP programming. The multi-objective optimization problem was transformed into a single objective optimization problem to the solver of a single objective optimization problem that can be performed. Genetic Algorithm was successful in finding the best solution for this scheduling in a month. Every testing has different time consumption to finding the best solution. We found the best solution in second testing needs 285.406s to calculate with the minimum function value is 2. The genetic algorithm is the right solution in helping to arrange difficult and time-consuming scheduling if done manually. Testing results are also affected by the value of probability crossover and mutation. Based on our experiments, the Binary GA can be used to support making the laboratory technicians scheduling in school/university/hospital in daily operation.

ACKNOWLEDGEMENT

The authors would like to express a sincere gratitude to the anonymous reviewers for their valuable comments and suggestions to improve the quality of this manuscript. In addition, the authors would also like to thank Bina Nusantara University for supporting this research project. This work is supported by the Directorate General of Strengthening for Research and Development, Ministry of Research, Technology, and Higher Education, Republic of Indonesia as a part of Applied Research entitled “Optimization for Multi-Objective Scheduling of Medical Personnel and Resources in Hospital Using Heuristic Approaches” with contract number: 036/VR.RTT/IV/2018 and contract date: 10 April 2018.

REFERENCES

1 S. Varghese, S. Kumar and R. Pooley, "Teacher information system using genetic algorithm," International Journal of Advanced Computational Engineering and Networking, vol. 4, no. 1, pp. 63-67, 2016.

2 A. T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, "Staff scheduling and rostering: a review of applications, methods, and models," European Journal of Operational Research, vol. 153, pp. 3-27, 2004.

3 E. I. Ásgeirsson, "Bridging the gap between self schedules and feasible schedules in staff scheduling," Annals of Operations Research, vol. 218, no. 1, pp. 51-69, 2014.

4 P. D. Causmaecker and G. V. Berghe, "Towards a reference model for timetabling and rostering," Annals of Operations Research, vol. 194, no. 1, pp. 167-176, 2012.

5 M. Gen and L. Lin, "Multiobjective genetic algorithm for scheduling problems in manufacturing systems," Industrial Engineering and Management Systems, vol. 11, no. 4, pp. 310-330, 2012.

6 A. Konak, D. W. Coit and A. E. Smith, "Multi-objective optimization using genetic algorithms: a tutorial," Reliability Engineering and System Safety, vol. 91, no. 9, pp. 992-1007, 2006.

7 A. Nanda, M. P. Pai and A. Gole, "An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach," International Journal of Machine Learning and Computing, vol. 2, no. 4, pp. 492-495, 2012.

8 A. Mardiyono, K. B. Seminar and H. Sukoco, "An intelligent system for course scheduling in higher educations," International Journal of Technology and Business Management, vol. 32, no. 1, pp. 29-34, 2014.

9 G. S. Budhi, K. Gunadi and D. A. Wibowo, "Genetic algorithm for scheduling courses," International Conference on Soft Computing Intelligence Systems and Information Technology, vol. 516, pp. 51-63, 2015.

10 L. G. Bharade, A. B. Khandekar and R. P. Bijwe, "Automatic time table generator using genetic algorithm," Imperial Journal of Interdisciplinary Research, vol. 2, pp. 132-135, 2016.

11 I. R. Ahmad, S. Sufahani, M. Ali and S. N. A. M. Razali, "A heuristics approach for classroom scheduling using genetic algorithm technique," Journal of Science and Technology, vol. 9, no. 3, pp. 10-15, 2017.

12 B. S. Kumar, G. Nagalakshmi and S. Kumaraguru, "A shift sequence for nurse scheduling using linear programming problem," IOSR Journal of Nursing and Health Science, vol. 3, no. 6, pp. 24-28, 2014.

13 M. Ito, M. Hirao and H. Hamahara, "A support system for nursery staff shift scheduling – a case study at a nursery school," Journal of Information Processing, vol. 26, pp. 294-300, 2018.

14 M. Labidi, M. Mrad, A. Gharbi and M. A. Louly, "Scheduling IT staff at a bank: a mathematical programming approach," The Scientific World Journal, vol. 2014, pp. 1-10, 2014.

15 S. S. S. Reddy, G. S. P. Reddy, P. V. Hemanth and P. Chatterjee, "Train time scheduling using genetic," International Journal of Civil Engineering and Technology, vol. 8, no. 12, pp. 410-413, 2017.

16 M. A. Al-Jarrah, A. A. Al-Sawalqah and S. F. Al-Hamdan, "Developing a course timetable system for academic departments using genetic algorithm," Jordanian Journal of Computers and Information Technology, vol. 3, no. 1, pp. 25-36, 2017.

17 A. Brezulianu, M. Fira and L. Fira, "A genetic algorithm approach for a constrained employee scheduling problem as applied to employees at mall type shops," International Journal of Advanced Science and Technology, vol. 14, pp. 1-14, 2010.

18 A. O. Adewumi and M. M. Ali, "A multi-level genetic algorithm for a multi-stage space allocation problem," Mathematical and Computer Modelling, vol. 51, no. 1-2, pp. 109-126, 2010.

19 S. K. Jha, "Exam timetabling problem using genetic algorithm," International Journal of Research in Engineering and Technology, vol. 3, no. 5, pp. 649-654, 2014.

20 D. Mittal, M. Sunasra and H. Doshi, "An art of generating automatic timetable using genetic algorithm," International Journal of Engineering Research and General Science, vol. 3, no. 2, pp. 842-848, 2015.

21 A. B. Senouci and S. A. Mubarak, "Multiobjective optimization model for scheduling of construction projects under extreme weather," Journal of Civil Engineering and Management, vol. 22, no. 3, pp. 373-381, 2016.

22 S. A. Zaki, A. A. A. Mousa, H. M. Geneedi and A. Elmekawy, "Efficient multiobjective genetic algorithm for solving transportation, assignment, and transshipment problems," Applied Mathematics, vol. 3, no. 1, pp. 92-99, 2012.

23 M. Othman, G. Gouw and N. Bhuiyan, "Workforce scheduling: a new model incorporating human factors," Journal of Industrial Engineering and Management, vol. 5, no. 2, pp. 259-284, 2012.

AUTHORS PROFILE

Antoni Wibowo has received his first degree of Applied Mathematics in 1995 and a master degree of Computer Science in 2000. In 2003, He was awarded a Japanese Government Scholarship (Monbukagakusho) to attend Master and PhD programs at Systems and Information Engineering in University of Tsukuba-Japan. He completed the second master degree in 2006 and PhD degree in 2009, respectively. His PhD research focused on machine learning, operations research, multivariate statistical analysis and mathematical programming, especially in developing nonlinear robust regressions using statistical learning theory. He has worked from 1997 to 2010 as a researcher in the Agency for the Assessment and Application of Technology – Indonesia. From April 2010 – September 2014, he worked as a senior lecturer in the Department of Computer Science - Faculty of Computing, and a researcher in the Operation Business Intelligence (OBI) Research Group, Universiti Teknologi Malaysia (UTM) – Malaysia. From October 2014 – October 2016, he was an Associate Professor at Department of Decision Sciences, School of Quantitative Sciences in Universiti Utara Malaysia (UUM). Dr. Eng. Wibowo is currently working at Binus Graduate Program (Master in Computer Science) in Bina Nusantara University-Indonesia as a Specialist Lecturer and continues his research activities in machine learning, optimization, operations research, multivariate data analysis, data mining, computational intelligence and artificial intelligence.

Filbert Ivander is a graduate in Computer Science Department, Bina Nusantara University, BINUS Graduate Program-Master in Computer Science, Jakarta, 10480, Indonesia.

Retrieval Number: D7204118419/2019©BEIESP
DOI:10.35940/ijrte.D7204.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)