Jurnal Mantik, 5 (3) (2021) 1799-1807
Published by: Institute of Computer Science (IOCS)
Jurnal Mantik
Journal homepage: www.iocscience.org/ejournal/index.php/mantik/index

Implementation of Genetic Algorithm in Making a Covid-19 Vaccination Schedule

K. Dilpani Laksmi[1], Bayu Angga Wijaya[2], Desniari Gea[3]

[123] Fakultas Teknologi dan Ilmu Komputer,
[123] Universitas Prima Indonesia, Jl. Sekip Simp. Sikambing, Kec. Medan Petisah, Kota Medan, Sumatera Utara 20111, Indonesia

E-mail : kdilpani@gmail.com[1], bayuanggawijaya@unprimdn.ac.id[2], desniarigea97@gmail.com[3]

ARTICLEINFO **ABSTRACT**

Article history :

Received: September 13, 2021
Revised: October 19, 2021
Accepted: September 02, 2021

Keywords:

Schedule,
Genetic Algorithm,
Desease,
Covid-19,
Vaccination

At the end of 2019, the World was attacked by a very deadly virus. Starting from the city of Wuhan in China, this virus began to grow and attack the world and became a pandemic. Until now the world is still being attacked by this pandemic. Various experts in this world took turns looking for a solution to this problem. Various studies and studies were carried out to find a way out and treatment. In the middle of 2020, research began to be intensively carried out to make a covid 19 vaccine. In doing things that are quite complicated, of course we need the help of technology to do it. In this case, we can use an algorithm in scheduling the covid 19 vaccination. In this regard, this study would like to discuss the application of genetic algorithms in making the covid 19 vaccination schedule. The genetic Algorithm was chosen because it can use many variables in its application compared to the FIFO method. The FIFO method is indeed simpler, the first to register will be scheduled, but to support more variables it is more efficient to use Genetic Algorithms. The use of genetic algorithms in this study was to find solutions and apply them to the program to schedule the covid 19 vaccination. genetics in scheduling accurate and effective COVID-19 vaccinations.

1. Introduction

At the end of 2019, the world was attacked by a very deadly virus. Starting from the city of Wuhan in China, this virus began to grow and attack the world and became a pandemic. Until now the world is still being attacked by this pandemic. Various experts in this world took turns looking for a solution to this problem. Various studies and studies were carried out to find a way out and treatment. In the middle of 2020, research began to be intensively carried out to make a covid 19 vaccine. The purpose of making a vaccine is to increase the human body's immunity to be able to survive the attack of the covid 19 viruses. At this time the vaccine has been made and is entering the distribution stage for the general public. In Indonesia, in particular, this is a challenge, because the Indonesian population is quite large, which is around 279 million people. Therefore, it is necessary to use a method that can support the distribution of vaccines appropriately and equitably for all Indonesian people.

In doing things that are quite complicated, of course we need the help of technology to do it. In this case, we can use an algorithm in scheduling the covid 19 vaccination. In this regard, this study would like to discuss the application of genetic algorithms in making the covid 19 vaccination schedule. John Holand from New York, United States first developed the Genetic Algorithm in a book entitled "Adaption in Natural and Artificial Systems" in 1975 (Puspita et al., 2016). This algorithm is part of an evolutionary algorithm that has an approach to Darwinian evolution in the fields of Biology such as inheritance, gene mutation, etc. The

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

technique in this genetic algorithm will search for solutions to get the best solution with specified criteria. Thus, genetic algorithms can be interpreted as a technique for optimizing the solution of a problem. Genetic algorithms are also grouped in metaheuristic algorithms because they include optimal search techniques (Ni Luh Gede Pivin et al., 2016).

The genetic Algorithm was chosen because it can use many variables in its application compared to the FIFO method. The FIFO method is indeed simpler, the first to register will be scheduled, but to support more variables it is more efficient to use Genetic Algorithms, in the design of this program, age grouping and place of residence are used. Genetic algorithms have properties that are formed from possible solutions to get the optimal solution to a problem. Optimization consists of various objects among the many solutions in the search space. The space point of each search has one feasible solution. The solution that is said to be feasible has a fitness value. In case of important findings for solving complete problems, good genetic algorithms are used for optimization. This study will project how to use genetic algorithms in finding solutions to a problem by applying them to a program to schedule covid 19 vaccination. "

2. Methods

This research methodology is divided into two parts, namely :

2.1 System Development Model

a. Analysis

At this initial stage, the researcher analyzes and creates a framework to start building the program. Adjust the program to be made according to needs. There are several parameters that will be entered, namely: Registration, Phase I Vaccination Scheduling, Phase I Vaccinations, Phase II Vasin Scheduling. The Phase 2 Vaccination Scheduling is carried out 28 days from the Phase I vaccination. The Genetic Algorithm used is expected to make the schedule accurate and effective.

b. Design

This stage is designing the user interface. Interface design aims to provide a visual description of the web-based application to be built, so that it will be easier to implement the application and will facilitate the creation of applications.

c. Coding

Coding is the process of translating designs and flowcharts into a programming language. The coding process in this study uses the PHP programming language.

d. Testing

The testing of this program is done by simulating the vaccination scheduling process. By entering all the required parameters. Then it will be seen whether there is an overlapping schedule or multiple scheduling for each registered person.

e. Maintenance

If at the time of testing errors and bugs are still found, then a re-evaluation will be carried out. Until the program can be run without any errors and bugs. If the program has no errors and bugs, then the program is ready to run in real-time.

2.2 Design Program

The work process on the application of genetic algorithms to the scheduling of the covid 19 vaccination can be described in the form of a flowchart program that is ready to run in real-time.

a. Flowchart

The steps and sequence of making software are then presented in the form of graphs and arrows called flowcharts. The flowchart is expected to show the flow of user interaction with the software better.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

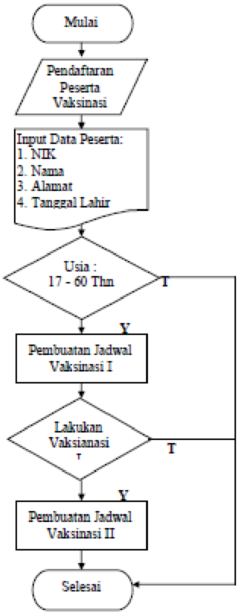

Fig. 1 Flowchart

3. Result and Analysis

3.1 Program View

a. Main menu

Fig. 2 Main Menu

b. Registration menu

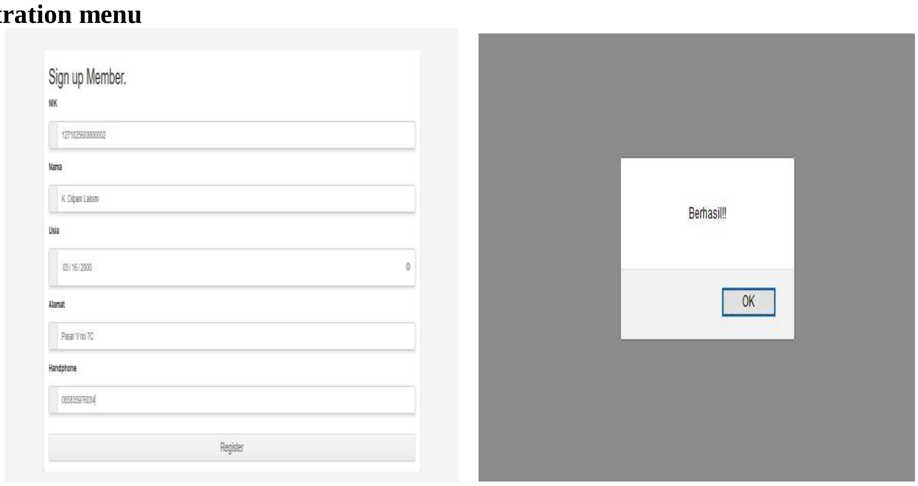

Fig. 3 Registration menu

c. Admin menu

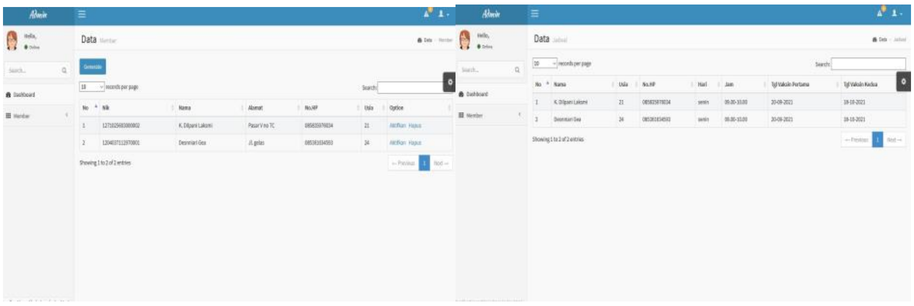

Fig. 4 Admin menu

3.2 Testing

Tests will be directly carried out in real-time on the application to see whether or not errors occur or whether there is still overlapping participant data. The time needed to vaccinate each person is about 3 minutes. Operational time for vaccination activities is 09.00 - 12.00 followed by 13.00 - 16.00. Thus the operational time in 1 day is 6 hours, then the probability of participants who can be vaccinated in 1 day: 60/3 minutes = 20 participants/hour. So in 1 day = 20 x 6 = 120 Participants, If in 1 day there are more than 120 participants who register, then the system will generate a schedule for the next day. If the next day the registration of new participants also exceeds the quota specified, the system will re-schedule to the next day, and so on. For more details, we can look at

Table 1

Registration of vaccination participants

Days	**Participation**
Monday	132
Tuesday	124
Wednesday	186
Thursday	123
Friday	139
Jumlah	704

With the number of participants who registered, it has exceeded the maximum daily quota of only 120 people. Under these circumstances, the algorithm used will automatically generate a schedule according to the specified parameters. We can see the results of generating a schedule in Table 2 below registration of vaccination participants.

Table 2

Registration Of Vaccination Participants

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09-Oct	Jan-20	121-140	241-260	361-380	481-500	601-620	
10-Nov	21-40	141-160	261-280	381-400	501-520	621-640	
11-Dec	41-60	161-180	281-300	401-420	521-540	641-660	
13-14	61-80	181-200	301-320	421-440	541-560	661-680	
14-15	81-100	201-220	321-340	441-460	561-580	681-700	
15-16	101-120	221-240	341-360	461-480	581-600	701-704	

a. Determination of Fitness Score

Before proceeding to the coding stage, you must first determine the fitness value. In this study, the fitness value can be obtained by the following calculations:

Fitness dilate is accentuated by two factors, namely

a = Schedule of Vaccination I carried out

b = Age of vaccination participants (18 - 60 years)

c = Place Capacity

d = Vaccination Schedule II

So that the calculation of the fitness value can be obtained as follows:

$$f = \frac{1}{1 + (a + b + c + d)}$$

Information:

a = Schedule of Vaccination I carried out
b = Age of vaccination participants (18 - 60 years)
c = Place Capacity
d = Vaccination Schedule II

The fitness with the greatest value (no conflict) is the best fitness, so the value should be 1. The fitness will be smaller if there are more clashes (clashes).

b. Number of chromosomes

Chromosomes are a collection of genes (a combination of lectures, space, and time) of several existing courses. In the beginning, the number of chromosomes that are generated is determined. One chromosome has one solution, so the more chromosomes that are generated, the more solution options (schedule) are generated, which will later choose the best one. The chromosomes will be stored in an array with the following format:

For example, the population value is 6
C[1] = [j;u;k;g]= [01;04;03;09]
C[2] = [j;u;k;g]= [20;05;07;01]
C [3] = [j;u;k;g]= [20;01;10;06]
C [4] = [j;u;k;g]= [10;04;03;04]
C [5] = [j;u;k;g]= [12;05;03;08]
C [6] = [j;u;k;g]= [02;01;08;03]
j= Schedule
u= Age
k= Capacity
g= Wave 2 Schedule
c= Chromosomes

It can be seen that Chromosome 1 has 12 genes, which means there are 12 vaccination participants. Each gene is a combination of Schedule, Age, Capacity, and Schedule II vaccines). The problem to be solved is the variable values of j, u, k, and g which must be the equation j+2u+3k+4g = 30, then the FO(Objective Function) that can be used is FO(Objective Function) of c = | (j+2u+3k+4g) – 30| . We calculate FO(Objective Function):

FO(Objective Function) c[1] = Abs((1 + 2*4 + 3*3 + 4*9) – 30)
= Abs((1 + 8 + 9 + 36) – 30)
= Abs(54 – 30)
= 24
FO(Objective Function) c[2] = Abs((20 + 2*5 + 3*7 + 4*1) – 30)
= Abs((20 + 10 + 21 + 4) – 30)
= Abs(55 – 30)
= 25
FO(Objective Function) c[3] = Abs((20 + 2*1 + 3*10 + 4*6) – 30)
= Abs((20 + 2 + 30 + 24) – 30)
= Abs(76 – 30)
= 46
FO(Objective Function) c[4] = Abs((10 + 2*4 + 3*3 + 4*4) – 30)
= Abs((10 + 8 + 9 + 16) – 30)
= Abs(43 – 30) = 13
FO(Objective Function) c[5] = Abs((12 + 2*5 + 3*3 + 4*8) – 30)
= Abs((12 + 10 + 9 + 32) – 30)
= Abs(63 – 30)
= 33
FO(Objective Function) c[6] = Abs((2 + 2*1 + 3*8 + 4*3) – 30)
= Abs((2 + 2 + 24 + 12) – 30)
= Abs(40 – 30)
= 10

The average of the objective functions is:
mean = (24+25+46+13+33+10)/6

= 25.167

c. Selection

Selection is the selection of which chromosomes will be used for the next algorithm process. Selection is determined based on the chromosome fitness value. The greater the fitness, the greater the chance of the chromosome to be selected. The method used for selection is the Roulette Wheel.

In carrying out the selection process, the fitness function = (1/(1+FO)), FO plus 1 is used to avoid the program experiencing errors due to the division of 0.

f[1] = 1 / (FO[1]+1)
= 1/24 + 1 = 0.0400
f[2] = 1 / (FO[2]+1)
= 1/25 + 1 = 0.0385
f[3] = 1 / (FO[3]+1)
= 1/46 + 1 = 0.0212
f [4] = 1 / (FO[4]+1)
= 1/13 + 1 = 0.0714
f[5] = 1 / (FO[5]+1)
= 1/33 + 1 = 0.0294
f [6] = 1 / (FO[6]+1)
= 1/10 + 1 = 0.0909
total_fitness = 0.04 + 0.0385 + 0.0212 + 0.0714 + 0.0294 + 0.0909
= 0.2914
Formula to find probability: P[i] = fitness[i] / total_fitness
P[1] = 0.04 / 0.2914
= 0.1373
P[2] = 0.0385 / 0.2914
= 0.1321
P[3] = 0.0212 / 0.2914
= 0.0728
P[4] = 0.0714 / 0.2914
= 0.2450
P[5] = 0.0294 / 0.2914
= 0.1009
P[6] = 0. 0909 / 0.2914
= 0.3119

From the probability it can be seen that C[6] has a large number of fitness then c[6] has the opportunity to be selected in the generation. The selection process uses the roulette wheel, therefore it must be obtained:

C[1] = 0.1009 + 0.3119 + 0.2450 + 0.0728 + 0.1373 = 0.8679
C[2] = 0.1009+ 0.3119 + 0.2450 + 0.0728 + 0.1373 + 0.1321 = 1
C[3] = 0.1009+ 0.3119 + 0.2450 + 0.0728 = 0.7306
C[4] = 0.1009+ 0.3119 + 0.2450 = 0.6578
C[5] = 0.1009
C[6] = 0.1009+ 0.3119 = 0.4128

The next selection process will be carried out using the roulette wheel after the cumulative is found. Random R in the range 0-1 will generate a number. If C[5] > R[k], chromosome 5 is selected as the parent, otherwise chromosome k is selected as the parent provided that C[k] > R > C[k-1]. The roulette wheel is rotated 6 times according to the number of populations and selects one chromosome in each new population:

R[1] = 0.398
R[2] = 0.501
R[3] = 0.822
R[4] = 0.009
R[5] = 0.201
R[6] = 0.284

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

As a chromosome in the new population, first R[3] is smaller than C[3] and larger than C[2] and C[2] will be selected, the result of the selection process with the new population is: c[1] = c[2]

c[2] = c[2]
c[3] = c[1]
c[4] = c[5]
c[5] = c[2]
c[6] = c[3]

The process of selecting a new chromosome:

c[1] = [02;01;08;03]
c[2] = [10;04;03;04]
c[3] = [01;04;03;09]
c [4] = [12;05;03;08]
c[5] = [02;01;08;03]
c [6] = [02;01;08;03]

d. Crossover

The crossover uses a method of randomly selecting positions in the parental chromosome and then exchanging genes, which is called a one-cut point, ie. The chromosomes selected by crossover will be used as crossover parameters (c).

Suppose we determine the probability of crossover is 25%, then it is expected that in one generation there are 50% of chromosomes (3 chromosomes) from one generation undergoing a crossover process. The process is as follows:

1. First generate a random value of R as many as the total population

R[1] = 0.159
R[2] = 0.340
R[3] = 0.006
R[4] = 0.760
R[5] = 0.191
R[6] = 0.259

2. The cut point is selected based on random numbers 1 to 3 with the number of crossovers available.

C[1] = 1
C[2] = 2
C[3] = 1
OS = Offspring
OS[5] = c [3] >< c [5]
=[01;04;03;09] >< [02;01;08;03]
= [02;04;03;09]
OS[3] = c[1] >< c[3]
= [02;01;08;03] >< [01;04;03;09]
= [01;01;08;03]
OS[1] = c[1] >< c[5]
= [02;01;08;03] >< [02;01;08;03]
= [02;01;08;03]

Thus the population becomes:

c[1] = [02;01;08;03]
c[2] = [10;04;03;04]
c[3] = [01;01;08;03]
c[4] = [12;05;03;08]
c[5] = [02;04;03;09]
c[6] = [02;01;08;03]

e. Mutation

In a population of chromosomes that undergo mutations based on the mutation rate parameter. By replacing a randomly selected gene, the mutation process gets a new value.

Genes = population * (total genes in chromosomes)
= 6 * 4

= 24
Determine pm 10% to 10% of the number of genes in the population:
mutations = 24 * 0.1
= 2.4 rounded to 2
Generated 2 and 5 as random numbers, the chromosome population will experience mutations:
c[1] = [02;05;08;03]
c[2] = [10;04;03;04]
c[3] = [01;01;08;03]
c[4] = [12;05;03;02]
c[5] = [02;04;03;09]
c[6] = [02;01;08;03]
The first mutation process is complete, we have completed one generation
c[1] = [02;05;08;03]
FO[1] = Abs((2 + 2*5 + 3*8 + 4*3) – 30)
= Abs((2 + 10 + 24 + 12) – 30)
= Abs(48 – 30)
= 18
c[2] = [10;04;03;04]
FO[2] = Abs((10 + 2*4 + 3*3 + 4*4) – 30)
= Abs((10 + 8 + 9 + 16) – 30)
= Abs(43 – 30) = 13
c[3] = [01;01;08;03]
FO[3] = Abs((1 + 2*1 + 3*8 + 4*3) – 30)
= Abs((1 + 2 + 24 + 12) – 30)
= Abs(39 – 30)
= 9
c [4] = [12;05;03;02]
FO[4] = Abs((12 + 2*5 + 3*3 + 4*2) – 30)
= Abs((12 + 10 + 9 + 8) – 30)
= Abs(39 – 30)
= 9
c[5] = [02;04;03;09]
FO[5] = Abs((2 + 2*4 + 3*3 + 4*9) – 30)
= Abs((2 + 8 + 9 + 36) – 30)
= Abs(55 – 30)
= 25
c[6] = [02;01;08;03]
FO[6] = Abs((2 + 2*1 + 3*8 + 4*3) – 30)
= Abs((2 + 2 + 24 + 12) – 30)
= Abs(40 – 30)
= 10
= 13
The average objective function after one generation is:
mean = (18+13+9+9+25+10) / 6
= 84 : 6
= 14

It can be seen that the resulting chromosomes in one gene are better than the previous gene. The next new chromosome is:

c[1] = [02;05;08;03]
c[2] = [10;04;03;04]
c[3] = [01;01;08;03]
c[4] = [12;05;03;02]
c[5] = [02;04;03;09]
c[6] = [02;01;08;03]

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

The chromosomes above will go through the same process continues until a predetermined generation.
Determined the best first 50 generations:
c= [01;03;05;02]
j=1 ; u=3 ; k=5 ; g=2
f = j+2u+3k+4g = 30
1 + (2*3) + (3*5) + (4*2) = 1 + 6 + 15 + 8 = 30

The results of the search for the problem to be solved are the values of the variables j, u, k, and g that meet the equation j+2u+3k+4g = 30, so that scheduling using genetic algorithms is declared fast and precise.

4. Conclusion

The use of the Genetic Algorithm in Scheduling the COVID-19 Vaccination is expected to speed up vaccination activities precisely and accurately. So that participants who have registered as soon as possible will get a schedule and be scheduled more than 1 time.

5. References

[1] Lesmana, N. I. (2017). Penjadwalan Produksi Untuk Meminimalkan Waktu Produksi Dengan Menggunakan Metode Branch and Bound.Jurnal Teknik Industri, 17(1), 42.
[2] Ni Luh Gede Pivin et al., 2016) Ni Luh Gede Pivin, S., I Made, S., & Suta, D. (2016). Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran. Journal of Applied Intelligent System, 1(3), 220–233.
[3] Paranduk, L., Indriani, A., Hafid, M., & Suprianto. (2018). Sistem Informasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Web. Seminar Nasional Aplikasi Teknologi Informasi (SNATi), E46–E50.
[4] Pranata, J., & Purba, K. R. (2016). Aplikasi Game Strategi Menggunakan Metode Algoritma Genetika untuk Pembuatan Pasukan. Jurnal Infra, 4(1), 199–204.
[5] Puspita, R. M., Arini, A., & Masrurah, S. U. (2016). Pengembangan Aplikasi Penjadwalan) Kegiatan Pelatihan Teknologi Informasi Dan Komunikasi Dengan Algoritma Genetika (Studi Kasus: Bprtik). Jurnal Online Informatika, 1(2), 76–81. https://doi.org/10.15575/join.v1i2.43
[6] Puspaningrum, W. A., Djunaidy, A., & Vinarti, R. A. (2013). Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. Jurnal Teknik Pomits, 2(1), 127–131.
[7] Patricia, E., & Suryono, H. (2015). Analisis Penjadwalan Kegiatan Produksi Pada Pt . Muliaglass Float Division Dengan Metode Forward Dan Backward. 43(1), 71–79.
[8] Pelajaran, M., Sdn, P., Cikarang, P., Kuliah, L., & Praktik, K. (2019). Perancangan sistem informasi penjadwalan mata pelajaran pada sdn pasirsari 01 cikarang. 1.
[9] Studi, G., Di, K., Dewanti, R., Novianingsih, K., & Agustina, F. (n.d.). MENGGUNAKAN ALGORITMA DEPARTEMEN PENDIDIKAN MATEMATIKA FPMIPA UPI). 43–53.
[10] Priandani, N. D., & Mahmudy, W. F. (2015). Optimasi Travelling Salesman Problem with Time Windows (Tsp-Tw) Pada Penjadwalan Paket Rute Wisata. Seminar Nasional Sistem Informasi Indonesia, November, 2–3.
[11] Sidin, U. S. (2016). Sistem informasi penjadwalan mata pelajaran berbasis web. SemanTIK, 2(1), 47–58.
[12] Soenandi, I. A. (2013). Konsep algoritma genetik biner untuk optimasi perencanaan jadwal kegiatan perkuliahan (. Jurnal Teknik Dan Ilmu Komputer, 02(07), 355–362.)
[13] Suhartono, E. (2015). Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang). Infokam, 2, 132–146.
[14] Suprayogi, D. A., & Mahmudy, W. F. (2015). Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry. Jurnal Buana Informatika, 6(2)
[15] Utama, D. M., Ardiansyah, L. R., & Garside, A. K. (2019). Penjadwalan Flow Shop untuk Meminimasi Total Tardiness Menggunakan Algoritma Cross Entropy–Algoritma Genetika. Jurnal Optimasi Sistem Industri, 18(2), 133. https://doi.org/10.25077/josi.v18.n2.p133-141.2019.