WS Similarity Check

el-15-07-13

Sand_102 -- No Repository 048

Document Details

Submission ID
trn:oid:::3117:483732102

Submission Date
Aug 15, 2025, 9:57 PM GMT+7

Download Date
Aug 15, 2025, 10:03 PM GMT+7

File Name
el-15-07-13.pdf

File Size
1.3 MB

12 Pages

4,116 Words

21,306 Characters

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Exclusions

- 11 Excluded Matches

Match Groups

12 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

11% Internet sources
0% Publications
2% Submitted works (Student Papers)

Match Groups

- **12** Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

11% Internet sources
0% Publications
2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Internet

www.icicel.org **11%**

ICIC Express Letters
Volume **15**, Number **7**, July **2021**

ICIC International ©2021 ISSN 1881-803X
pp. **791–802**

GENETIC ALGORITHM IMPLEMENTATION FOR APPLICATION OF SHIFTING WORK SCHEDULING SYSTEM

Wahyu Sardjono[1], Wowon Priatna[2], Dwi Sulistyo Nugroho[2]
Aninda Rahmasari[3] and Erma Lusia[4]

[1]Information Systems Management Department
BINUS Graduate Program – Master of Information Systems Management
[3]Scientific Research Publication, Research & Development
[4]Research Capacity Building, Research & Development
Bina Nusantara University
JL. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
wahyu.s@binus.ac.id; {aninda.rahmasari; ermalusia}@binus.edu

[2]Informatics Engineering
Bhayangkara University
Jl. Harsono RM No. 67 Ragunan Pasar Minggu, South Jakarta, DKI Jakarta 12140, Indonesia
wowon.priatna@dsn.ubharajaya.ac.id; dwi.sulistio16@mhs.ubharajaya.ac.id

Received October 2020; accepted January 2021

Abstract. *Companies that have irregular work schedules must apply work shifts to their employees. To meet a large volume of work with a small number of employees, it is necessary to schedule work shifts that need to be implemented for a certain period of time and with minimum cost planning while considering a number of restrictions related to labor performance constraints. Currently, many companies in determining work shift scheduling are still carried out conventionally manually using only Microsoft Office with the Excel program, which results in frequent shift schedules that conflict with other employees' shift schedules and require a long time or result in various obstacles. To solve this problem, work shift scheduling can be done by implementing a genetic algorithm so that the shift scheduling problem is more regular and does not clash with the schedule of other employees' work shifts. The results of the calculation of the genetic algorithm are used to design a shifting scheduling application, the application is modeled using the unified modeling language, and this is intended so that system development can be done optimally with not too many human resources. The results obtained from the implementation of the calculation of the genetic algorithm using the shift time variable, day variable, shift employee name variable did not find any penalties. So these results can be applied to the company as an optimal solution for scheduling work shifts.*
Keywords: Genetic algorithm, Unified modeling language, Scheduling, Work shifting

1. **Introduction.** Company organizations that have irregular work schedules must apply work shifts to their employees. To meet a large volume of work with a small number of employees work shift scheduling needs to be implemented over a minimum period of time and cost planning while considering a number of restrictions related to labor constraints. Many company organizations in making work shift schedules still use a manual system only using Excel so that the impact in scheduling is that shift schedules often clash with other employees and take a long time or cause all kinds of problems. One of the suitable methods for solving scheduling problems is genetic algorithm [1,2].

The objectives of this study are

1) to solve problems in work shift scheduling using genetic algorithms;
2) designing a work shift scheduling application based on the calculation of the genetic algorithm.

DOI: 10.24507/icicel.15.07.791

The results of this study are expected to solve problems in scheduling work shifts more regularly, no schedule conflicts with other employees and having a scheduling application that will make it easier for companies to make work shift schedules. Genetic algorithm is an optimization algorithm for different problems [3]. Genetic algorithm is also a stochastic search based on the principles of natural selection and recombination [4] and generates selection to estimate solutions to solve problems [5]. Genetic algorithm performance depends on coding and operator selection [6]. In order to solve a more complicated problem, an integrated genetic algorithm uses hybridization methods to increase the effectiveness of its performance [7]. In compiling a genetic algorithm into a program, several stages of the process are needed, namely the process of making the initial generation, the genetic process, the selection process, and the repetition of the process [8]. Figure 1 describes the cycle genetic algorithm introduced by David Goldberg [9].

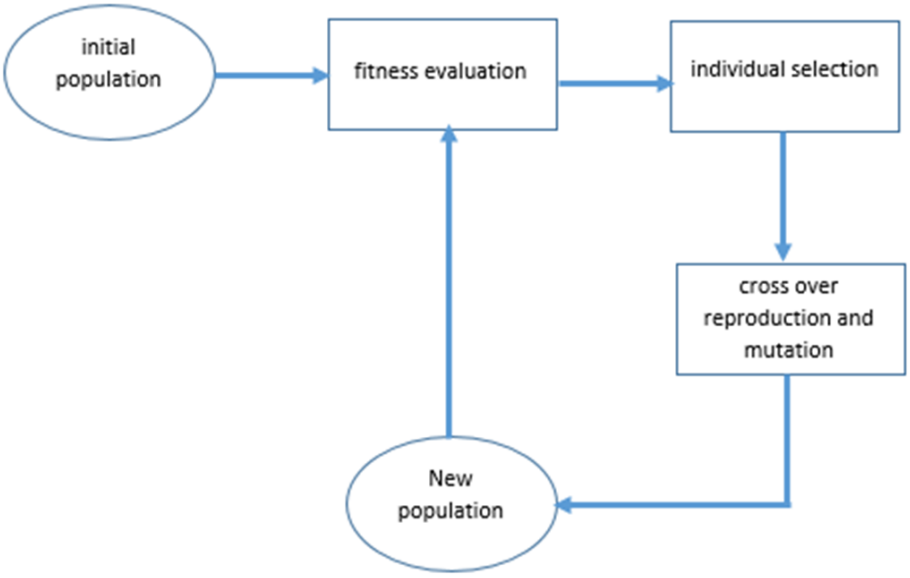


FIGURE 1. Life cycle of genetic algorithms [9]

2. Literature Review.

2.1. **Definition of algorithm.** An algorithm is a sequence of steps to solve a problem. Algorithms are at the heart of computer science or informatics. Many branches of computer science are referred to in algorithmic terminology. Algorithm is a logical and systematic arrangement to solve a problem or to achieve certain goals [10]. So it can be concluded that an algorithm is a logical and systematic step to solve a problem. However, do not assume that algorithms are always synonymous with computer science. In general, those who work on the process are called processors. Processors can be humans, computers, robots, mechanical devices, electronic devices and others. Implementing the algorithm means working on the steps written in the algorithm.

2.2. **Basic algorithm structure.** Algorithm contains steps to solve a problem. These steps can be actions or events, selection of actions and repetition of actions. Here are the three basic structures of the algorithm, namely:

1) Sequence Structure
   The sequence structure is the basic structure of the algorithm in which the instructions will be executed sequentially, used for programs whose instructions are sequential.
2) Selection Structure
   A branching structure is the basic structure of an algorithm where instructions or statements will be executed if they need or fulfill a condition.

3) Repetition Structure
   The loop structure is the basic structure of the algorithm where instructions will be executed repeatedly if they fulfill it or not fulfill a condition.

2.3. **Genetic algorithms.** The use of algorithms that mimic the workings of living things in solving problems – optimization problems has been introduced since the 1960s, which is commonly known as computational evolution – is genetic algorithms. The implementation of genetic algorithms for optimization problems is a very interesting subject to study, this is because genetic algorithms are very powerful and are an optimization technique that can work on many problems that are very difficult to solve with conventional techniques [11].

The genetic algorithm is a numerical optimization algorithm that is inspired by natural selection and natural genetics [12]. The genetic algorithm is divided into several parts, namely [13]:

1) *Allele*, a value that is in a combination of schedules.
2) *Genotype*, a value that states the base unit which is forming a certain meaning in a unit called chromosome.
3) *Chromosome*, a combination of genes that form a particular value.
4) *Individual,* is a value or situation that states one possible solution and the problem is raised.
5) *Population*, is a group of individuals to be processed together in the evolutionary cycle.
6) *Fitness value*, states the value of how good the value of an individual or the solution is.
7) *Selection*, a process to get chromosomes good quality in the next generation.
8) *Crossover,* a method of marrying two chromosomes parent (parents) with the aim of getting a better child.
9) *Mutation,* the process of changing the arrangement of genes in a chromosome to make a difference in the population.

2.4. **Scheduling.** According to [14], scheduling can be defined as the process of allocating resources to work on a set of tasks within a certain period of time with 2 important meanings as follows.

1) Scheduling is a decision-making function to create or determine a schedule.
2) Scheduling is a theory that contains a set of basic principles, models, techniques, and logical conclusions in the decision-making process that provides an understanding of the scheduling function.

Scheduling is a process of organizing, selecting, and determining the time to use existing resources to produce output as expected in the expected time as well [15]. It can be concluded that scheduling is a strategic part of the planning process and setting work sequences and allocating resources both time and facilities for each operation that must be completed. On allocating resources there are important goals that the scheduling process will achieve. There are two targets to be achieved through scheduling, namely the number of outputs produced and the due date. These two targets are stated through scheduling criteria such as minimum makespan (total time spent in the production process), minimum mean flow time (average production process time), minimum mean lateness (average delay), minimum tardiness of delay, minimum mean tardiness (average delay), and minimizing the number of tardy (number of delays). In general, scheduling is a planning activity to determine when and where each operation as part of the overall work must be performed on limited resources.

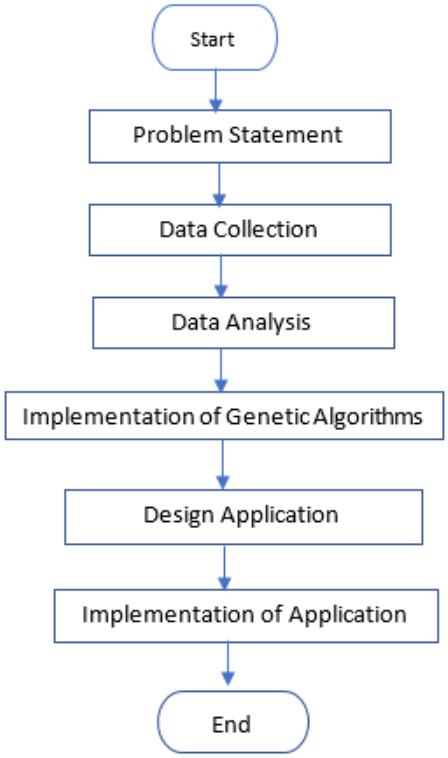


FIGURE 2. Research design

3. **Methodology.** The stages to be carried out in this research can be seen in Figure 2.

3.1. **Problem identification.** Problem identification is taken based on the main problems that exist in the research background based on the system that runs in the scheduling in the company.

3.2. **Data collection techniques.** In this study use a case study at PT. XYZ, where to retrieve data is done through observation and direct interviews with managers, staff and employees to determine the conditions of current work schedule scheduling.

3.3. **Data analysis methods.** Based on the data collection that has been done, it is found

1) Users involved in scheduling the shifting work system are the HRD Division, customer care, NOC and TOC;
2) In the process of making and preparing a shifting work system schedule requires employee data, detailed data on shifting working hours, detailed data on shifting work days.

3.4. **Implementation of genetic algorithms.** The flow of algorithm implementation to be carried out in this study is depicted in a flowchart which can be seen in Figure 3.

Based on Figure 3, the stages of implementing the genetic algorithm are

1) Initial value determination is to determine the size of a random individual obtained from a population that depends on the problem at hand.
2) Reproduction is to determine the chromosomes from the selection results.
3) Reproduction stage is to determine the suitability between individuals and populations using the fitness function. Within each generation the most suitable individual is selected and many techniques can be used [16]. To find the fitness value, it can be used in Equation (1).

$$\text{Fitness} = 1/(1 + (\text{penalty1} + \text{penalty2} + \text{penalty3} + \text{penalty4})) \quad (1)$$

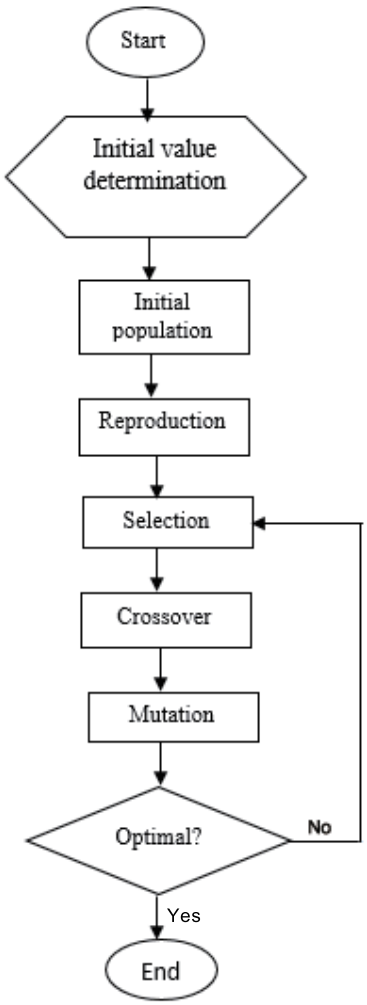


FIGURE 3. The flow of the genetic algorithm implementation

4) Selection is a process to screen the chromosomes better by selecting from two chromosomes randomly rather than to get the crossover and mutation values [17]. To make a selection, it is done by finding the percentage of the fitness value divided by the total fitness multiplied by 100%.

$$\text{Fitness Percentage} = \text{fitness}/(\text{total fitness}) \times 100\% \quad (2)$$

5) Crossover is the stage randomly selecting one position from the parent chromosome to be exchanged for genes [16] which will get the better chromosome [18,19].
6) The mutation in the process refers to this probability parameter [20] to change the value of a particular gene from 0 to 1 or vice versa. This mutation process will produce new individuals who are from the combination of the parent chromosomes [21].

3.5. **Application design.** After the application design stage is complete, proceed with creating an application based on the results of application design where the programming language uses PHP and data storage uses the MySQL database.

4. Result and Discussion.

4.1. Genetic algorithms.

4.1.1. *Raising the initial population.* In this stage, it is to determine the shift time variable, day variable, employee name variable, then it is used as a chromosome. The initial population was formed based on the number of N chromosomes. Then it is generated

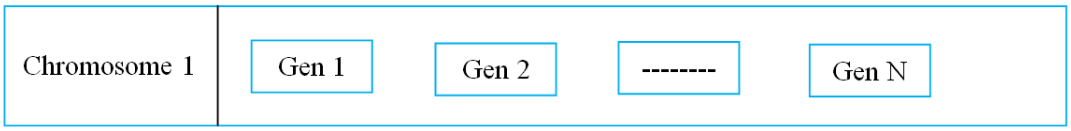


FIGURE 4. Initial population

randomly for the chromosome evaluation process, where the length of the genes consists of N genes, namely gene 1, gene 2 to gene N as shown in Figure 4.

4.1.2. *Chromosome formation.* Chromosome formation defines the number of genes used and can represent the solution to the problem. In determining a solution, problems that arise can be represented by a set of parameters. This parameter is called a gene that unites to form a chromosome, the number of available work days is 1 week, where the working day starts from Monday to Sunday, the working time in a day is 24 hours starting from 08.00 AM to 08.00 AM the following day, the number of chromosomes is 20, while one chromosome consists of N genes. The coding that is done is permutation coding, which is the formation of a chromosome that is formed by a number sequence that states the numbers in a sequence. In determining a gene, it is done by randomizing its length value, the value of the gene is 0, 1, 2, 3, where 0 is the off shift, 1 morning shift, 2 day shift and 3 is the night shift. Figure 5 shows how to get the chromosome value.

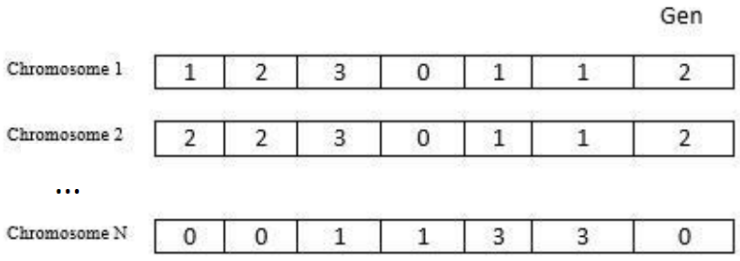


FIGURE 5. Process of getting chromosome values

4.1.3. *Fitness.* To get a fitness score, you must first determine the penalty rules:

a) There was no off in one week.
b) There are off times more than 2 days in 1 week.
c) The operator must have a minimum break of 8 hours.
d) There are no female operators on night shifts.

Here are some chromosomes whose fitness is calculated, as shown in Figure 6.

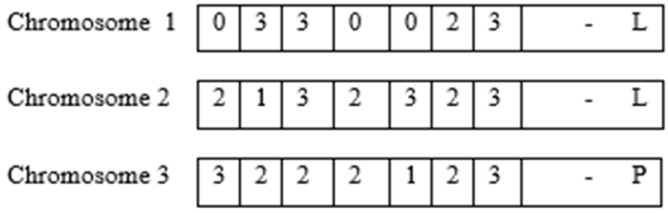


FIGURE 6. Fitness chromosomes

From the 3 chromosomes above produce:
No penalty off in one week = 0
Penalty off more than 2 days = 1
Penalty for rest less than 8 hours = 0

Female employee penalty = 0
Calculate the fitness of chromosome 1: $1/(1+1) = 0.5$

No penalty off in one week = 1
Penalty off more than 2 days = 0
Penalty for rest less than 8 hours = 2
Female employee penalty = 0
Calculate the fitness of chromosome 2: $1/(1+3) = 0.25$

No penalty off in one week = 1
Penalty off more than 2 days = 0
Penalty for rest less than 8 hours = 1
Female employee penalty = 2
Calculate the fitness of chromosome 3: $1/(1+3) = 0.25$

4.1.4. *Selection.* In the selection process the greater the fitness value, the more likely it is to be selected as the parent. The fitness percentage value based on Equation (2) can be stated in Table 1.

Table 1. Fitness percentage

Individual	Fitness	Percentage
1	0.50	52.63%
2	0.25	26.31%
3	0.25	21.05%
Total	**1.00**	**99.99%**

Table 1 shows the probability level of individual selection for the selection process. Individual 1 has the greatest probability, namely 52.63% being selected as the first parent for the formation of new offspring in individual selection for the chromosome crossover process. And individual 2 with a probability of 26.31% is selected as the second parent for the formation of new offspring in the selection of individuals for the chromosome crossover process.

4.1.5. *Crossover.* In this process, select two parent chromosomes to be cross-bred randomly. The cross breeding process uses a one point crossover. In this process, child chromosomes will be produced as a result of the cross-breeding of the two parents, where this child chromosome contains genes from the combination of the two parents and is different from the parent genes. The following is an example of the cross-breeding process (crossover), as shown in Figure 7.

4.1.6. *Mutations.* Mutations are carried out randomly by selecting or modifying genes on chromosomes. Mutation results are obtained from low fitness values or vice versa [20], as seen in Figure 8.

4.2. **Application.** The application design is done by using two diagrams from UML (Unified Modeling Language), namely use case diagrams and class diagrams. Use cases are used to model the interaction between actors and the system, while the class diagram model application is in the form of classes that map the pages connected to the work shift scheduling application.

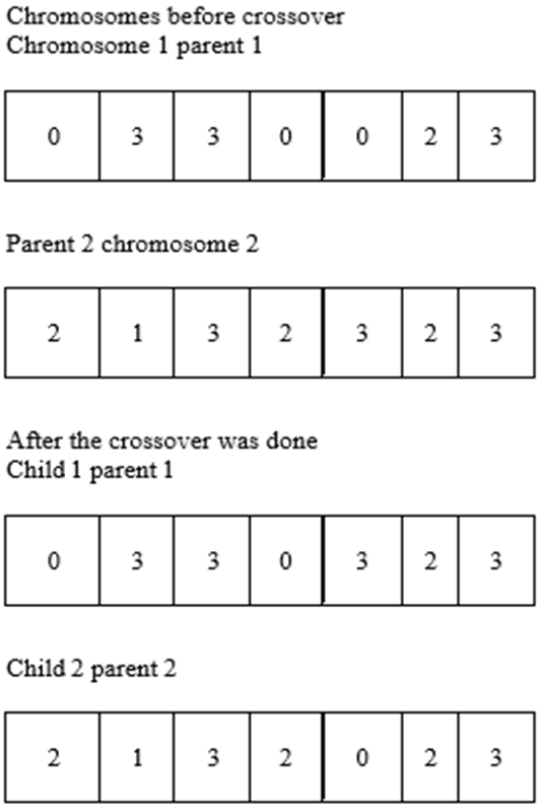


FIGURE 7. Cross-breeding process

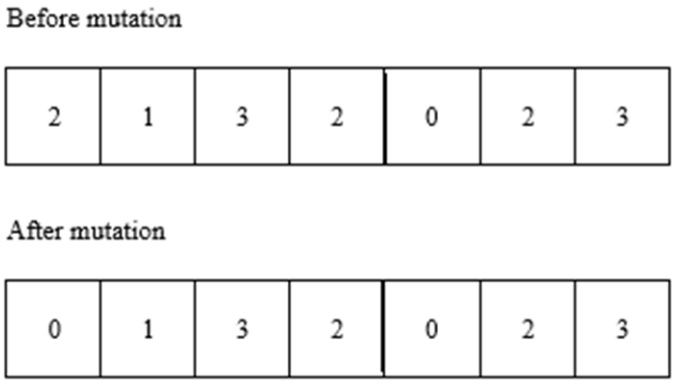


FIGURE 8. Mutation results

4.2.1. *Use case diagram.* The use case in Figure 9 is an application modeling for work shifting scheduling applications where there are 3 actors, namely staff, supervisors and Human Resource Development (HRD). Staff can only see schedules, while supervisors are tasked with making a shifting work schedule which is a reference for working hours for employees in their respective divisions, and can have access to view shift data and employee data in their respective divisions, while HRD has authority to approve the schedule submitted by the supervisor and have access to manage employee data and applicable shifting detail data.

4.2.2. *Class diagram.* The class diagram in Figure 10 begins with the login class which is implemented by the *tbl_user* class because it is logged in, the *tbl_user* class requires the *tbl_employee* class because the user is an employee of the company. The *tbl_scheduling* class also depends on the *tbl_employee* class because the employees of the company see, create and approve the schedule. *Class_division* and *tbl_division* are the composition of

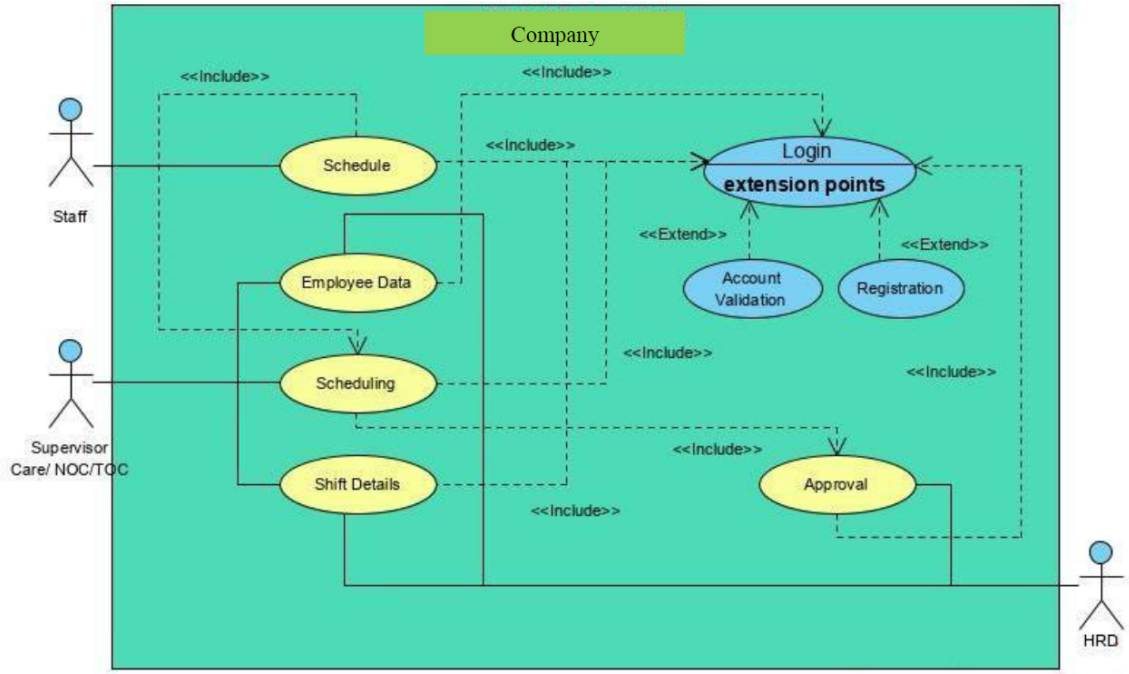
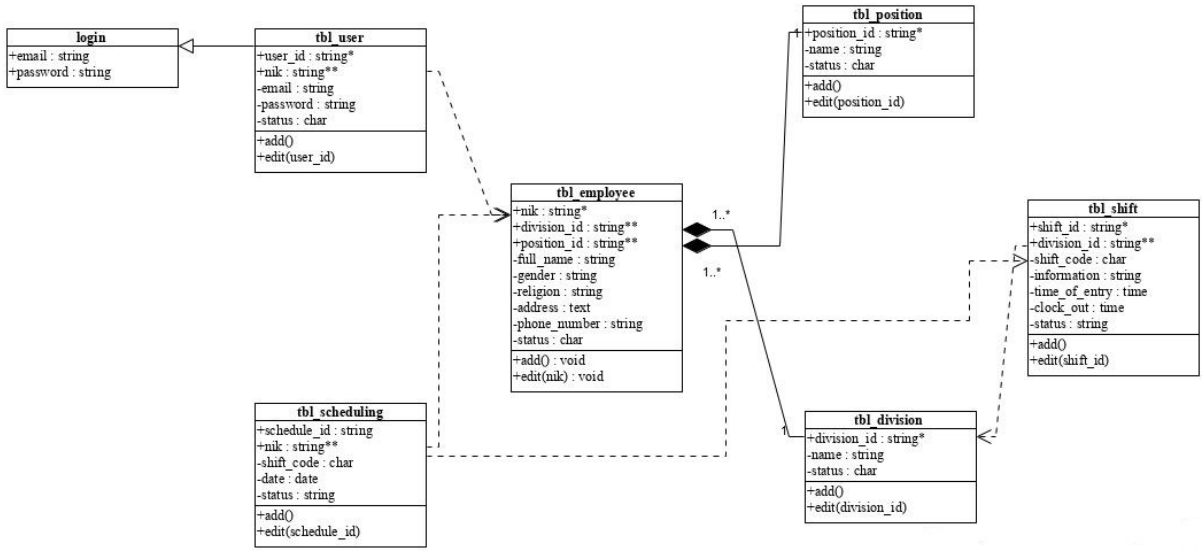


FIGURE 9. Use case diagram



* is first primary key; ** is second primary key

FIGURE 10. Class diagram

tbl_employee, meaning that the class is part of the *employee_class*. Whereas *tbl_scheduling* depends on the *tbl_shift* class because if there is no shift class, scheduling cannot be done and the *tbl_shift* class depends on the *tbl_division* class because the data shift class is from the *tbl_division* class.

4.3. **Application implementation.** The results of UML modeling are then made an application using the PHP Codeigniter Framework and data storage using Mysql. Here are some views of the work shifting scheduling application.

4.3.1. *Detail shifting.* Shift detail view can be seen after logging in from the browser by the user and selecting the shift detail menu on the main menu. The shifting detail page can be seen in Figure 11.

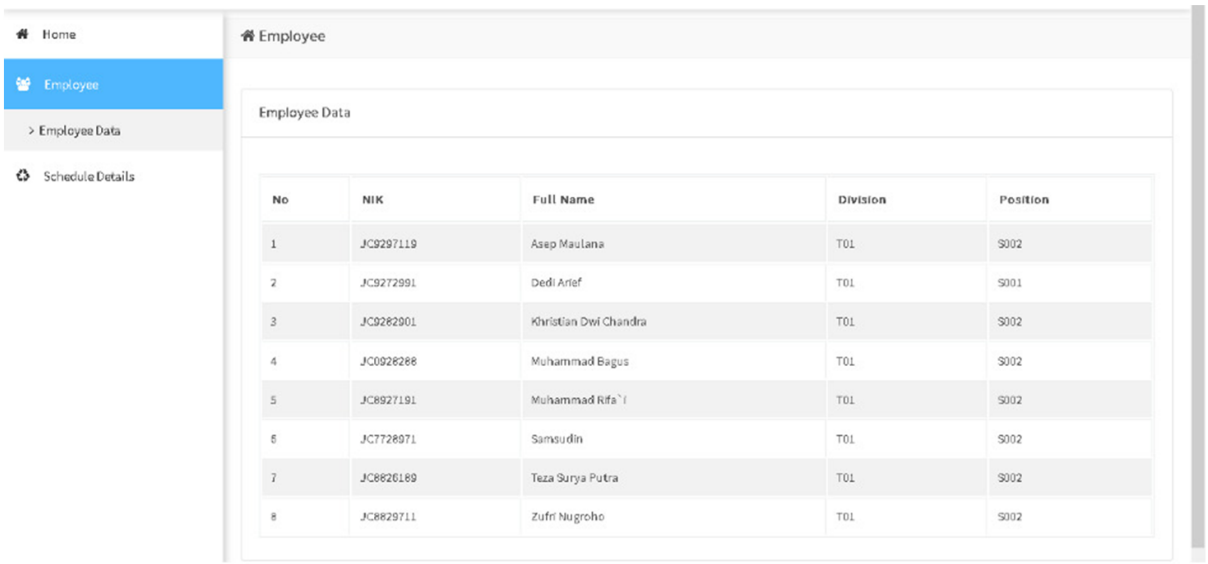


FIGURE 11. Detail shifting

4.3.2. *Creating a schedule.* Activities create a schedule that can be accessed by the supervisor whose function is to generate schedules. When saved, this page will immediately provide notification to the HRD on the appearance of the HRD approval. The display of making a schedule can be seen in Figure 12.

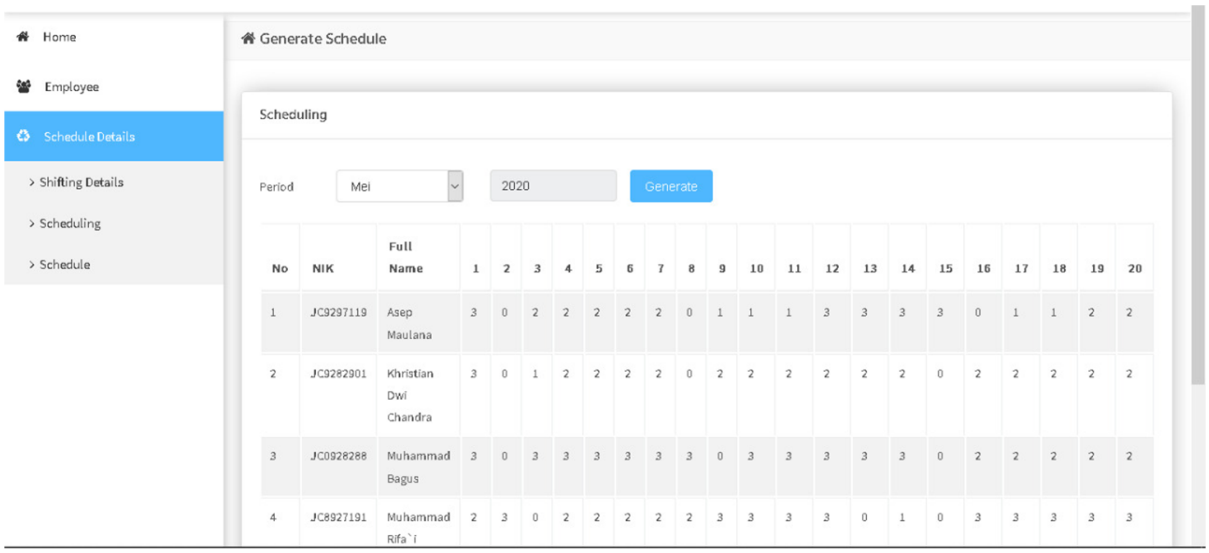


FIGURE 12. Generating a shifting schedule

4.4. **Implementation of genetic algorithms in applications.** The application of genetic algorithms is located on the scheduling form submenu from the Schedule Details menu. This submenu will display the employee data that has been sorted based on their respective employee divisions. The supervisor will enter the schedule period data you want to make along with the details of the year. After all data input for the period and year have been entered, the new genetic algorithm that has been built will work. The input data that has been entered will be an additional parameter where scheduling will be carried out by taking the variable number of employees, variable employee identification number, variable number of days, variable shift code and variable period proposed.

4.5. **Results of shift scheduling by system.** Based on the shift scheduling carried out by the system with the employee data of PT. XYZ, a solution search will be carried out until it gets the optimal value with the reference value 0. Then the schedule obtained is in the form of a random value with a range of 0 to 3, where it is the coding type of shift applied to this application which will be the reference for employee schedules for the next month.

5. **Conclusion.** Based on the research and discussion carried out, it can be concluded:

1) The shifting scheduling aims to be able to do scheduling automatically, thus displaying the day schedule for each employee.
2) To maximize employee scheduling using genetic algorithms, parameters are needed as a reference such as the number of employees, employee data.
3) The implementation of genetic algorithms in work shift scheduling displays several forms consisting of employee data, division data, shift detail data, shift scheduling.
4) The implementation of genetic algorithms in work shift scheduling is obtained optimally where there are no clashes between employee shift schedules and for female employee shifts, night work shifts are not found as calculated in the genetic algorithm process.

Further research related to the application of genetic algorithms for shift work scheduling can be continued by looking at gender in particular, because the treatment between female and male workers should be different between working hours per day and each month.

REFERENCES

[1] D. Kannan, K. Bajpayee and S. Roy, Solving timetable scheduling problems using genetic algorithm, *Int. J. Recent Technol. Eng.*, vol.7, no.5, pp.168-170, 2019.

[2] A. I. Diveev and O. V. Bobr, Variational genetic algorithm for NP-hard scheduling problem solution, *Procedia Computer Sci.*, vol.103, pp.52-58, 2017.

[3] S. M. Elsayed, R. A. Sarker and D. L. Essam, A new genetic algorithm for solving optimization problems, *Engineering Application Artificial Intelligence*, vol.27, pp.57-69, 2014.

[4] J. Luo and D. El Baz, A survey on parallel genetic algorithms for shop scheduling problems, *Proc. of 2018 IEEE 32nd Int. Parallel Distrib. Process. Symp. Work. (IPDPSW2018)*, pp.629-636, 2018.

[5] Y. Tamerabet, F. Adjadj and T. Bentrcia, Evaluation of the genetic algorithm performance for the optimization of the grand potential in the cluster variation method, *Calphad*, vol.61, pp.157-164, 2016.

[6] J. Magalhães-Mendes, A comparative study of crossover operators for genetic algorithms to solve job scheduling problem, *WSEAS Transactions on Computers*, vol.12, no.4, 2013.

[7] L. Asadzadeh, A local search genetic algorithm for the job shop scheduling problem with intelligent agents, *Comput. Ind. Eng.*, vol.85, pp.376-383, 2015.

[8] E. Tozzo, A. P. Costa and I. D. Lins, A hybrid multi-objective genetic algorithm for scheduling heterogeneous workover rigs on onshore oil fields, *Journal of Petroleum Science and Engineering*, vol.195, 2020.

[9] K. Han, R. Zuo and D. Zhang, Application of a genetic algorithm to groundwater pollution source identification, *Journal of Hydrology*, vol.589, 2020.

[10] D. Kristiadi and R. Hartanto, Genetic algorithm for lecturing schedule optimization, *Indonesian Journal Computer Cyber System*, vol.13, no.1, p.83, 2019.

[11] S. Zolfaghari, V. Quan, A. El-Bouri and M. Khashayardoust, Application of a genetic algorithm to staff scheduling in retail sector, *Int. J. Ind. Syst. Eng.*, vol.5, no.1, pp.20-47, 2010.

[12] Kinast, K. F. Doerner and S. Rinderle-Ma, Biased random-key genetic algorithm for cobot assignment in an assembly/disassembly job shop scheduling problem, *Procedia Computer Science*, vol.180, pp.328-337, 2021.

[13] D. Rooyani and F. M. Defersha, An efficient two-stage genetic algorithm for flexible job-shop scheduling, *IFAC-Papers OnLine*, vol.52, no.13, pp.2519-2524, 2019.

[14] B. Limwanich and R. Wongsathan, Efficiency improvement of job scheduling by using genetic algorithm: A case study in electronic industry, *IEEE International Conference Engineering and Management*, pp.1750-1754, 2011.

[15] Shukla, Anupriya, H. M. Pandey and D. Mehrotra, Comparative review of selection techniques in genetic algorithm, *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, Greater Noida, India, 2015.
[16] N. A. Fountas and N. M. Vaxevanidis, Optimization of fused deposition modeling process using a virus-evolutionary genetic algorithm, *Computers in Industry*, vol.125, 2021.
[17] S. Noor, M. I. Lali and M. S. Nawaz, Solving job shop scheduling problem with genetic algorithm, *Sci. Int. (Lahore)*, vol.27, no.4, pp.3367-3371, 2015.
[18] P. M. R. Mahalakshmi, Flexible employee scheduling with multi skill training program using genetic algorithm, *Int. J. Res. Eng. Appl. Manag.*, vol.3, no.9, pp.53-58, 2017.
[19] J. Wang and K. Chu, An application of genetic algorithms for the flexible job-shop scheduling problem, *International Journal of Advancements in Computing Technology*, vol.4, no.3, 2012.
[20] F. Werner, A survey of genetic algorithms for shop scheduling problems, *Math. Res. Summ.*, vol.2, p.15, 2017.
[21] A. Sukstrienwong, A genetic-algorithm approach for forming optimized location-based group of member, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2229-2245, 2019.