PAPER • OPEN ACCESS

Modified genetic algorithm for employee work shifts scheduling optimization

View the article online for updates and enhancements.

Modified genetic algorithm for employee work shifts scheduling optimization

N W S Saraswati[1], I D M D Artakusuma[2] and I G A A D Indradewi[3]

[1,2,3] Study Program of Informatics Engineering, STMIK STIKOM Indonesia, Denpasar-Bali 80225, Indonesia

E-mail: sumartini.saraswati@stiki-indonesia.ac.id [1], dewadwi@hotmail.com [2], didyrakata@gmail.com[3]

Abstract. Arranging an employee shift work's schedule requires high accuracy. It is because we have to pay attention to several constraints simultaneously. The genetic algorithm presents as a method which can automatize the process of arranging the schedule as well as optimizing the result of the schedule. The Shala Bali is a hospitality business whose scheduling was complicated because of the number of the employes. This research aimed at producing a shift work schedule of the employees in a week and to know the optimum genetic algorithm parameter in this case. The constraints that were taken into account in the arrangement of the schedule included the schedule conflict of the employees in one shift, schedule conflict of employees in 1 day, the same composition of employees per shift, employees should not get morning shifts after being in night shift the night before, each shift has at least 1 employee in the front office, and each employee is required to get 1 day off within 1 schedule period. This study was able to produce an optimal work schedule of employees with crossover probability (Pc) of 0.6, and mutation probability (Pm) of 0.3. The modification algorithm in chromosome generation and chromosome structure in this study results that changes in gene length (additional number of employees) do not have to be followed by an increase in the number of chromosome populations to get optimum results.

1. Introduction

Artificial Intelligence methods are expected to be problem solvers in solving real-world problems that can be represented in data structures. Genetic algorithms (GAs) as one of the techniques in artificial intelligence are quite recent techniques of optimization, whose basic concept is mimicking the evolution of a species, according to the Darwinian theory of the 'survival of the fittest' [1]. Evolutionary programming techniques, based on genetic algorithms, can be applied to many hard optimization problems, such as function optimization with linear and nonlinear constraints, travelling salesman problems, scheduling, partitioning, and control problems [2]. Genetic Algorithms have become increasingly popular for solving complex optimization problems such as those found in the field of scheduling or timetabling [3]. Because scheduling problems are generally NP-hard and are often optimized by heuristics alone, they have made good targets for GA research and much progress has been made in using GA for scheduling optimization [4]. The importance of this technique is still developing because evolutionary programs are parallel in nature, and parallelism is one of the most promising directives in computer science [2].

Multi-objective scheduling by the Genetic Algorithm describes methods for developing multi-objective solutions for modelling general production scheduling equations in the literature as flow hops,

job shops, and open shops. Multi-objective flow hops, job shops, and open shops are all highly relevant model in manufacturing, class scheduling or automotive assembly [5]. The problem of scheduling employee shifts is similar to the problem of scheduling classrooms in lectures. Therefore in this study genetic algorithms were chosen to solve the work shift scheduling of The Shala Bali employees. We used the case in Shala Bali to illustrate that the genetic algorithm can solve the optimization of employee work shift scheduling well.

As a growing hospitality business, The Shala Bali employees a lot of employees, thus the scheduling is complex. The management of the work schedule in The Shala Bali is managed by the manager manually, that is by typing the schedule with the help of Microsoft Excel with the consideration that each employee has equal shift work. Every schedule period lasts for one week, and in one day, four work shifts are available, namely: morning, afternoon, evening, and night. The types of employees' jobs are divided into two; those are front office and back office.

The management of work schedule as described above has some shortcomings, including long hours working schedule where employee may have a continuous work schedule that last for 16 hours, and the numbers of the day off per week for each employee are often not met, thus making the work distribution among employees uneven. In this study, 12 work schedule periods of the employee were collected and obsserved, from which found 5 mistakes in one period of the employee's work schedule.

To answer this problem, this study aimed at producing an optimum schedule of an employee's shift work in a week and identifying the optimum genetic algorithm parameter in this case

2. Methodology and Related Work

2.1. Previous Research of Genetic Algorithm
In many industries, work must be done 24 hours a day, 7 days a week. The typical work schedule in this context consists of a weekly cycle or repeated cycle for several weeks. Work is commonly divided into different shifts, namely: morning shift (D), evening shift (E), and night shift (N) [6]. This work division can commonly be found in a service business, for example in the field of hospital and hospitality services.

In one research, scheduling shift work was done for the nurses in the hospital. Assigning nurses into time slot arranged in such a way so a shift work schedule of the nurses was produced to fulfill all requirements of scheduling. A schedule solution was produced with a four-week time slot duration. The constraints which could be avoided in this nurse shift work schedules are a hard limitation and soft limitation. Hard limitation included the availability of a certain number of nurses in certain shifts and a nurse should work every third weekend, etc. A soft limitation included avoiding overtime hours, avoiding three consecutive days of working, etc. A schedule solution in a genetic algorithm was represented by a chromosome [7]. In this study, two kinds of chromosomes were being used, the first was the traditional bit-string structure to represent every schedule. The second solution was using two-dimensional array chromosome structures to represent every schedule. The results of previous study showed that the scheduling with two-dimensional chromosomes gave better results than scheduling with a bit-string chromosome [8].

Previous studies in this topic mostly take the issue of scheduling the shift work of nurses from the different objective functions. In this study, the objectives of scheduling were aimed to produce a schedule that simultaneously minimizes the hospital staff's housing expense and equalize the total of overtime hour, other than maintaining the high service level. To obtain an optimal nurse schedule with the number of treatment gains from the simulation model, a genetic algorithm (GA) was used with two-point crossover and random mutation. After running the algorithm, the expense and the total of nurses between the existing nurse schedule and those made with GA were compared. For January 2013, the nurse scheduling obtained by GA could save 12% in employment cost per month and 13% in the number of nurses if compared to existed schedule, by still maintaining equal number distribution of employee's overtime [9].

The similar scheduling can also be done to make a schedule of shift work of doctors in the Pediatric Department of Prince Sultan Military Medical City (PSMMC) in Riyadh, United Emirate Arab. Genetic Algorithm approach useing bit-cost matrix where every cell showed the existence of a breach of constraints. The trial result showed that the suggested method produced faster doctors' schedule and with reduction of constraints of violated number compared with traditional manual method [10].

In the educational field, we need scheduling with GA to schedule lectures or teaching. This scheduling form can be a single scheduling form or multi-objective scheduling form. Single scheduling is represented by research for scheduling company tutor [4]. Meanwhile, multi-objective scheduling is represented by research of lectures scheduling where this scheduling has several variables such as courses, lecture, students, lecturing room and time slot [11–13].

Generally, in scheduling using a genetic algorithm, scheduling resources are static data. However, in reality, scheduling resources are most often dynamic data, as proven by previous ttudies about the use of a genetic algorithm in accomplishing dynamic scheduling [14–16].

In the case of employee work shifts scheduling at Shala Bali, the Genetic Algorithm scheduling process was carried out dynamically based on the number of employees that can potentially change. In this research, the chromosome form was able to accommodate this dynamic scheduling.

This is the novelty of the present research, where prior research has not mentioned special treatment related to computational simple Genetic Algorithm, which is done for scheduling cases. In this study, the addition of chromosome repair methods in computation were used to determine the success in obtaining the optimum solution.

2.2. Data Collection

Data obtained from observation and interview results with The Shala Bali side produce several rules that can be considered in the making of the employee schedule. The Shala Bali uses weekly scheduling that consists of 7 workdays, 1 day divided into 4 shifts that are morning from 07.00 – 15.00 WITA, afternoon from 11.00 – 19.00 WITA, evening from 15.00 – 23.00 WITA, and night from 23.00 – 07.00 WITA. In one shift work, there is a minimal one front-office employee. The employee of The Shala Bali has right for 1 day-off in 1 week, and also if the employee has gotten night shift, then it is not allowed to get a morning shift the next day.

2.3. Constraint Determination

Based on the result of data collection about the rule of the scheduling of employee shift work in The Shala Bali then the scheduling constraints will be determined as follows. The constraints will be divided into two forms: namely: hard constraints and soft constraints. Hard constraints are constraints that should be fulfilled into one shift work, meanwhile soft constraints are constraints whose violations can be minimalized. Here are the constraints:

- Hard Constraints
  - There are no similar persons in one shift work.
  - In one shift work, there is a minimal one front-office employee.
  - An employee should get the same total shift in every scheduling period and get once day-off.
- Soft Constraints
  - There is are no similar persons in one workday.
  - If there is an employee get a night shift, then it is not allowed to get a morning shift in the next day.

2.4. Chromosome Representation

Chromosome generation in a genetic algorithm is done in random. Chromosome states one possible alternative solution. Population size depends on the problem that will be solved. In this study the chromosome length was 28 gens according to the number of shifts work available in a week, as shown in Table 1.

In this experiment, it used 21 employees to be scheduled in a week. Thus, in one shift work consists of 4 or 5 employees. This amount obtained from this calculation:

$$N = \frac{Total\ Employee\ \times Total\ Workdays\ for\ Each\ Employee}{Total\ Shift} \quad (1)$$

If calculated, the result obtained is 4,286 which means there is a slot that consists of 4 employees and there is a slot that consists of 5 employees. To get the total slot that consists of 5 employees, we did modulus functions with an equation:

$$N = (Total\ Employee\ \times Total\ Workdays\ for\ Each\ Employee)\ mod\ Total\ Shift \quad (2)$$

For the data above, it was obtained 8 time slot that consists of 5 employees. As simplification, the chosen time slot is shifted 1 to 8.

The chromosome was built with 28 genes according to the total shift. Each gene contained employee code as much as 5 employee code in gene 1 to 8 and 4 employee code in gene 9 to 28, as shown in Table 2.

Table 1. Time Slot Representation.

Shift	Day						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Morning	1	2	3	4	5	6	7
Afternoon	8	9	10	11	12	13	14
Evening	15	16	17	18	19	20	21
Night	22	23	24	25	26	27	28

Table 2. Chromosome Representation.

Gene 1	Gene 2	....	Gene 28
K01 K07	K06 K05	......	K19 K17
K03 K02	K03 K01		K16 K14

2.5. Chromosome Correction

Chromosome correction is an effort done to avoid the breach of hard constraints that is the total of workdays of an employee. In this study, each employee was required to work for six shiftss in one week so that each chromosome generated was first made sure whether each employee has appeared as many as six times. If there are employees' IDs that appeared less or more than six times, then replacement algorithm had to be applied.

The replacement algorithm is performed by tracing a chromosome and counting the number of placement of each employee on that chromosome. If the number of employee placements in 1 chromosome is less than 6, which means that the employee was assigned less than 6 shifts a week, the employee code was added in the chromosome for the number of deficiencies. If the number of employee placements in 1 chromosome was more than 6, then it was reduced to 6. The replacement positions was carried out randomly.

2.6. Fitness Function

A chromosome is evaluated based on certain function as performance measurement. In a natural evolution, an individual that has high fitness value will survive. Meanwhile, an individual that has low fitness value will die. The value resulted from the function indicates how optimal the obtained solution. The value that are resulted from the fitness function also represents how many amounts of requirements that are violated, so that in the case of employee scheduling, the lower amount of violation produced, the better the solution produced will be. For each violation that occurs, it would be given a value of 1. To avoid an infinite fitness value then the total amount of all violation was added 1. To minimize the occurrence of violation in hard constraints, the total violation included the in hard constraint was multiplied by 0.8 and the violation in soft constraint was multiplied by 0.3 so that one violation in hard constraint worth more than twice the soft constraint violation.

$$fitness = \frac{1}{1 + \left((0.8 \times \sum BS) + (0.3 \times \sum BH) + (0.3 \times \sum BMP) + (0.8 \times \sum JK) + (0.8 \times \sum KK)\right)} \quad (3)$$

Explanation:
BS = Clash in one shift work (*Hard Constraint)*
JK = Total workdays each employee should be six times in one shift period (*Hard Constraint)*
KK = Total employee composition in one shift (*Hard Constraint)*
BH = Employee that work more than one shift per day (*Soft Constraint)*
BMP = Employee get night shift and get morning shift in the next day (*Soft Constraint)*

2.7. Crossover

Crossover is used as a chromosome cutting method randomly and a combination of the first part of parent chromosome 1 and the second part of parent chromosome 2 and vice versa. Crossover can only be done if a random number generated for chromosome is less than crossover probability (Pc) that is determined by the user. To do chromosome cutting then it needs a cutting point. This cutting point will be generated randomly and has a range value from 1 to n-1, where *n* is the length of the chromosome. This study used a single-point crossover where the cutting point is generated from a random number of 1 to 27.

2.8. Mutation

The offspring chromosome produced from the process of the crossover will be continued with the process of mutation. The purpose of mutation means changing one or more genes from the chromosome. In this study, the mutation was done in reciprocal exchange mutation, so there would be randomly 2 bit-genes chosen in one chromosome to be exchanged.

Probability for the mutation to happen depending on the mutation probability (Pm), mutation will only happen if the generation of the random number smaller than Pm (Mutation Probability). Mutation probability that is too small would trap it in optimum local, and mutation probability that is too high would cause difficulty in achieving convergence.

3. Experiment Result

The experiment is done to know whether the genetic algorithm can solve the problem of employee shift work scheduling of The Shala Bali and to know how chromosome correction can make the modified algorithm solve the dynamic of employee's number.

The hardware specifications used for the experiment were:
- Processor intel core i3 3217u 1.8GHz
- RAM : 1x8GB DDR3 1600MHz
- VGA : Intel HD Graphic 4000
- HDD 500GB

And the softwares required for the experiment were:
- Windows 7 ultimate 64bit
- Visual Studio 2010
- Net framework 4.0
- MySQL 5.0

3.1. The Testing of Genetic Algorithm on Pc Variable

From the result of the testing above, it can be seen that the different values of Pc (Crossover Probability) Variable affected the result of the genetic algorithm. In this case, the optimum value of Pc (Crossover Probability) is on 0.6 value.

Table 3. Experiment Result with Crossover Probability Variable.

No	Population	Generation	PC	PM	*Fitness*
1	25	1000	0.5	0.3	0.1524
2	25	1000	0.6	0.3	0.1530
3	25	1000	0.7	0.3	0.1374
4	25	1000	0.8	0.3	0.1458
5	25	1000	0.9	0.3	0.1390

3.2. The Testing of Genetic Algorithm on Pm Variable

From the result of the testing above, it can be seen that the difference of Pm (Mutation Probability) variable value affected the result of the genetic algorithm. In this case, the optimum changes of Pm (Mutation Probability) value are in 0.3 value.

Table 4. Result of Experiment with Mutation Probability Variable.

No	Population	Generation	PC	PM	*Fitness*
1	25	1000	0.5	0.1	0.14402
2	25	1000	0.5	0.2	0.14490
3	25	1000	0.5	0.3	0.15116
4	25	1000	0.5	0.4	0.13670
5	25	1000	0.5	0.5	0.14384

From the result above, it can be seen that the optimum value of each variable that determines the final value of fitness as shown in Table 5.

Table 5. Variable Optimum.

Variable's Name	Optimum Value
Population	15
Crossover Probability(Pc)	0.6
Mutation Probability (Pm)	0.3

3.3. Testing by The Changing of Number of Employees

Testing was done by changing the number of employees that are scheduled to know the effect of population size on the result of optimum scheduling as shown in table 6 and table 7.

It is interesting to observe how the chromosome modification triggered the changes in the gene's length that represented the changes number of employees, which was followed by the changes in population size, but it did not affect the fitness value, as shown on data in Table 6. The optimum result was sufficiently fulfilled by the larger number of a generation when the number of employees increases.

Table 6. Testing Result of The Changing of Number of Employees with The Changing Population Size.

Number of Employees	Population Size	Fitness	Computational Time	Total Generation
20	15	1	00:01:48.60	13092
40	35	1	00:07:46.600	16080
60	55	1	00:34:28.250	18987
80	75	1	01:35:51.150	30360

Based on the experimental results in Table 7, it can be observed that optimal results were still obtained even though the number of employees increases significantly.

Table 7. Testing Result of The Changing of Number of Employees with Fixed Population Size.

Number of Employees	Population Size	Fitness	Computational Time	Total Generation
20	15	1	00:01:34.50	11263
40	15	1	00:05:29.750	16175
60	15	1	00:10:36.45	12066
80	15	1	00:19:04.080	19651

Based on Table 6 and 7, it can be concluded that an increase in population size and increase in number of employees only affects the increased computation time because it is accompanied by an increase in the number of generations to achieve the optimum solution.

4. Conclusion

On a real case at the Shala hotel, scheduling 6 front office employees and 15 back office employees, the mutation probability value is 0.3 and the crossover probability 0.6 provides the optimum value. This optimum value means that the schedule is formed without any clash of all predefined constraints.

Chromosome modification and gene replacement processes when crossed-over to ensure an even distribution of employee work shifts causes changes in the number of employees (changes in chromosome length), but these changes were not necessarily followed by changes in population numbers to maintain optimal schedule results. This contrasts with the common logic that when we extend the

chromosome size, we need to increase the population size to provide sufficient variation of the solution in the initial population.

5. Future Research
This study has limitations in applying Genetic Algorithms to a real-world case. For further research, it can be examined how the genetic algorithm is designed to solve work shift scheduling cases that can be applied in general to all cases.

References
[1]   García-Martínez C, Rodriguez F J and Lozano M 2018 Genetic algorithms *Handbook of Heuristics*
[2]   Rocke D M and Michalewicz Z 2006 Genetic Algorithms + Data Structures = Evolution Programs *Journal of the American Statistical Association*
[3]   Aickelin U and Dowsland K A 2004 An indirect genetic algorithm for a nurse-scheduling problem *Comput. Oper. Res.* **31** 761–78
[4]   Greenspan R 2005 A Multi-objective Genetic Algorithm for Employee Scheduling *Univ. Illinois* 1–8
[5]   Bagchi T P 2011 *Multiobjective Scheduling by Genetic Algorithms*
[6]   Granfeldt C 2016 Using a Genetic Algorithm to Solve the Rotating Workforce Scheduling Problem *Linkoping Univ.*
[7]   Omara F A and Arafa M M 2010 Genetic algorithms for task scheduling problem *J. Parallel Distrib. Comput.* **70** 13–22
[8]   Dean J 2008 Staff Scheduling by a Genetic Algorithm with a Two-Dimensional Chromosome Structure *Proc. 7th Int. Conf. ...* 1–15
[9]   Leksakul K and Phetsawat S 2014 Nurse scheduling using genetic algorithm *Math. Probl. Eng.* **2014** 0–17
[10]   Alharbi A and Alqahtani K 2016 A Genetic Algorithm Solution for the Doctor Scheduling Problem *The Tenth International Conference on Advanced Engineering Computing and Applications in Sciences* pp 91–7
[11]   Kristiadi D and Hartanto R 2019 Genetic Algorithm for lecturing schedule optimization *IJCCS (Indonesian J. Comput. Cybern. Syst.* **13** 83
[12]   Parera S, Sukmana H T and Wardhani L K 2016 Application of genetic algorithm for class scheduling (Case study: Faculty of science and technology UIN Jakarta) *Proc. 2016 4th Int. Conf. Cyber IT Serv. Manag. CITSM 2016* 3–7
[13]   Mardiyono A 2014 an Intelligent System for Course Scheduling in Higher Educations *Int. J. Inf. Technol. Bus. Manag.* **29** 29–34
[14]   Eletrotécnica E 2017 *Dynamic Scheduling for Maintenance Tasks Allocation supported by Genetic Algorithms*
[15]   A.Kaleeswaran, V.Ramasamy and P.Vivekanandan 2013 Dynamic Scheduling Of Data Using Genetic Algorithm In Cloud Computing *Int. J. Adv. Eng.* **5** 327–34
[16]   Madureira A, Ramos C and Silva S do C 2003 Using Genetic Algorithms for Dynamic Scheduling *Proc. Prod. Oper. Manag. 2003 Conf. (POMS'2003), Georg. (EUA).*