Minimization Total Tardiness in Lathe and Turret Department Using Genetic Algorithm

I Gede Wira Ranata¹, Dida Diah Damayanti², Murni Dwi Astuti³
Industrial Engineering, Faculty of Industrial Engineering
Universitas Telkom, Bandung, Indonesia
¹wiraranata@student.telkomuniversity.ac.id, ²didiah@telkomuniversity.ac.id, ³murnidwiastuti@telkomuniversity.ac.id

Abstract. In recent years the development of manufacture in Indonesia has increase rapidly. Many companies must improve their performance to meet the requirement of the customer needs. Scheduling is an important phase in production. One kind of the problem is scheduling in parallel machine. Total jobs will be processed in the paper are 249 and it divided into 3 types of machine. Every job has their due dates and the problem is some jobs exceed the due dates and make tardiness in some jobs. The total tardiness in the existing scheduling are 8134 hours. This paper set a proposed approach using genetic algorithm to minimization total tardiness in real case factory with a parallel machine type scheduling. The proposed method compared with dispatching heuristic rule EDD and the existing scheduling of the factory. The proposed approach results demonstrate that a genetic algorithm give a better scheduling solution then the EDD rule and the existing schedule. Proposed scheduling using GA can reduce the tardiness becomes 723.97 hours or 91.10% from the existing schedule.

1. Introduction
Improving performance in now days is very important for the factory. One kind of classical and important problem is scheduling in parallel machine. Scheduling have an important role in manufacturing system. Scheduling is a process of making a decision about the allocation of resource to processed in some facilities of factory [1]. In parallel machine problem, consists of more than one (n) that must be processed in more than one (m) parallel machine [2]. The purpose of parallel machine is sequencing and make an optimal assignment job to every machine [3]. In this case, there are 3 type of machine consist of 7 units machine lathe type I, 7 units machine lathe type II, and 5 units machine turret. The amount of jobs that processed to every machine are 95 jobs in lathe machine type I, 99 jobs in lathe type II, and 55 jobs in machine turret. Every job has their due dates, with the due date the company have a problem. The problem is there are some jobs can’t finish on time. That condition made tardiness of some jobs. The total amount tardiness are 8134 hours with total tardy job are 129. The problem caused by the existing method of the scheduling can’t give an optimal scheduling result. The existing method of the companies is using a random method. Certainly, this condition will give a disadvantage for the company, the company can get a penalty cost for the tardiness. To determine the delivery dates to customer is an important phase because if the company promising delivery due to far from the order it can make the satisfy of the customer will decrease [4]. In the real case, parallel machine scheduling with objective to minimization tardiness can include to be a NP-hard problem [5]. Using an exact method to solve tardiness problem in parallel machine scheduling with large scale of job will need
large amount of time and memory in computational result, so that make using exact method not effective [6]. Based on the problem and that constraint, this paper aimed to minimization of tardiness in lathe and turret department. This study proposed a meta-heuristic genetic algorithm to minimization the amount of tardiness in processing jobs.

2. Literature Review

2.1 Definition of Parallel Machine Scheduling

There are so many types of machine scheduling, one of them is parallel machine. In this type of scheduling there are more than one machines available and every machine could use to process some activities [7]. The process of scheduling, involve a process to make component and it called by job [8]. To sequencing n jobs on m parallel processors, there are some objective values one kind of them is tardiness minimization. Parallel machine scheduling has been got an important attention to improve the performance. Dispatching rule EDD is one kind of method to minimize tardiness in parallel machine scheduling. Below are the steps EDD rule [9]:

- a. Sort all jobs based on the smallest due date.
- b. Allocate each job to every machine with a smallest due date get first priority.
- c. Loading of every machine must analysed for the next allocated jobs, the other job has not been allocated to the machine will be allocated to the smallest machine loading.

2.2 Genetic Algorithm

Genetic algorithm is problem solving method in the scheduling problem based on the principle of outgrowth and inheritable. This method starts with an initial population and it was made randomly to search the optimal solution and make the best solution to solve the problem [10]. Genetic algorithm consists of 5 components, the components are genetic representation, a way to create and determine the initial population, parameter to evaluate the fitness value, genetic operator, and value of every parameter [11]. GA start with generates a population. This population will be the possible solution to solve the problem [12]. In recent years, some scheduling problem in parallel machine scheduling had been done using GA. Minimization total tardiness in widely problem using genetic algorithm also produce a good solution. It prove by the result of solve 250 problems and can reduce the tardiness for 93.60%, solve 1125 can reduce the tardiness for 84,04% and 74,67% [13]. The application of GA when comparison with heuristic method in tardiness minimization make a better result for 3,45% [14]. In another research, proposed method using GA show the relationship between tardiness and the amount of scheduling problem, this research suggested to use much iteration because can give a better solution [15]. GA also comparison with dispatching heuristic rule. The result is GA better than that dispatching heuristic rule. Scheduling using genetic algorithm can reduce the tardiness for 84,79% [16]. When GA comparison with branch and bound method, solving the problem using GA is faster than using that method. GA can speed up the computational time for 87,35% [17]. The other benefit from the proposed method is can reduce the number of tardy jobs so can make the product deliver on time to the customer [18].

2.3 Step of GA

Scheduling using genetic algorithm consist of some steps, below are step of using genetic algorithm in scheduling problem [19]:

- a. Problem representation, this is the first step using GA, one kind of the objective function is minimization tardiness [11].
- b. Make an initial population is the next step, initial population will be the input in the systems of GA. Using a heuristic method will give advantages because give a faster solution and better solution [20].
- c. Evaluate fitness value, the function is to determine the potential chromosome to result a better solution [11].
- d. Crossover, process of produce a new solution (child) and it called by offspring. There is a parameter in crossover which is called by crossover probability (Pc) [21]. Crossover process
need two chosen chromosome and it called by parent to produce offspring with expectation can result a better fitness value from the old chromosome [18].

e. Mutation, the function of mutation is to prevent the chromosome loss of their gen. Same with crossover, mutation also have a parameter which is called by mutation probability (\(P_m\)) [21].

f. Survivor selection function to choose the best chromosome from the GA process after the crossover and mutation process so can make a better solution. If the new chromosome produces smaller tardiness than the old chromosome it will be the new solution but, if the old chromosome gave a smaller tardiness, the old chromosome will be the best solution [21].

g. Steady state is a condition to stop the scheduling using GA, the criteria to find the steady state if the fitness value from new chromosome worse than the current chromosome, the tardiness never produce a better solution from the current tardiness, and the GA find the maximum iteration [21].

3. Finding

3.1 Existing Scheduling

Total amount of tardiness using the existing scheduling are 8134 hours and consist of 129 tardy jobs. Below is the tardiness in every type of machine:

| No | Machine Type    | Tardiness (Hours) | Amount of Tardy Jobs |
|----|----------------|-------------------|----------------------|
| 1  | Lathe Type I   | 2568              | 51                   |
| 2  | Lathe Type II  | 2776              | 48                   |
| 3  | Turret         | 2790              | 30                   |

3.2 EDD Dispatching Rule

This paper using EDD rule to make an initial solution to be the input of GA. EDD rule can also use to minimizing tardiness on \(m\) parallel machines [9]. Using EDD rule can give a better solution in computational time need to determine the initial population using genetic algorithm [20].

3.3 Scheduling Using Genetic Algorithm

3.3.1 Representation Problem

Scheduling with genetic algorithm start with the objective function, the objective function is to minimization total tardiness. Below is the mathematical model that will used in this research [13]:

a. Mathematical model to calculate the completion time of total jobs:

\[
ST_{j(1)}=0 \quad (1)
\]

\[
CT_{j(i)} = ST_{j(i)} + PT_{j(i)} \quad (2)
\]

\[
ST_{j(i)}=CT_{j(i-1)} \quad ; i>1 \quad (3)
\]

\[
CT_{j(i)} = ST_{j(i)} + PT_{j(i)} \quad ; i>1 \quad (4)
\]

b. The tardiness of every job can be calculated by this mathematical model:

\[
T_{j(i)} = \max (CT_{j(i)}-DT_{j(i)};0) \quad (5)
\]

c. To calculate the total tardiness of parallel machine scheduling, below is the mathematical model:

\[
T = \sum_{M=1}^{m} \sum_{j=1}^{J} T_{j(i)} \quad (6)
\]

d. Job can only be processed one time and only in one machine, below is the mathematical model:
The machine only can process one job in same time, below is the mathematical model:

\[ \sum_{m=1}^{\text{Mc}} \sum_{j=1}^{\text{indenj}} A_{(m,j,z)} = 1, \quad (7) \]

f. The value of variable \( A_{(m,j,z)} \) can be determine by:

\[ A_{m,j,z} = \begin{cases} 1, & \text{job } j \text{ will be processed in machine } m \\ 0, & \text{Otherwise} \end{cases} \quad (9) \]

Explanation:
- \( Mc \) = index machine, \( Mc = 1, 2, \ldots, m \); where \( m \) is amount of machines
- \( j_i \) = index job, \( j_i = 1, 2, \ldots, j_i \); where \( j \) is the amount of jobs
- \( PT_{j_i} \) = processing time every job
- \( ST_{j_i} \) = Start time every job
- \( CT_{j_i} \) = completion time every job
- \( DT_{j_i} \) = due date every job
- \( T_{j_i} \) = tardiness every job

3.3.2 Initial Population and Set the Parameter

This paper used EDD rule to make the initial population. The other population will be randomly generated by the systems. Set the parameter of GA can see below:

a. The population size of this research is 65 \[ 13 \].
b. Crossover probability is 0.65 \[ 13 \].
c. Mutation probability is 0.05 \[ 13 \]
d. Maximum iteration is 100 \[ 22 \] \n
3.3.3 Evaluate fitness value

Below is the example of calculate fitness value from some of the initial population

| Tardiness | Fitness |
|-----------|---------|
| Chromosome 1 | 108 | 0.00926 |
| Chromosome 2 | 163 | 0.00613 |
| Chromosome 3 | 163 | 0.00613 |
| Chromosome 4 | 73 | 0.0137 |

Based on Table 2, the best fitness value is in chromosome 4, that indicates chromosome 4 can produce the smallest tardiness from the other chromosome.

3.3.4 Crossover

The process of crossover is generated a random number through the application. The value of random number is from 0 – 1.

| Chromosome | Crossover Probability |
|------------|-----------------------|
| 1 | 0.206502876 |
| 2 | 0.433281221 |
| 3 | 0.678711584 |
| 4 | 0.805888088 |
Based on the generated random number through the systems. The crossover probability was set to 0.65, chromosome 1 and chromosome 2 will be crossover because the crossover probability of those chromosome is smaller than 0.65. Those chromosomes were chosen in hope can produce a smaller tardiness than before so can make a new solution. Below is the result of crossover from chromosome 1 and chromosome 2:

| Chromosome 1 | 7 | 8 | 3 | 4 | 6 | 0 | 2 | 1 | 5 | 9 |
| Chromosome 2 | 0 | 2 | 5 | 3 | 6 | 9 | 4 | 1 | 7 | 8 |
| **Sequence Swap** | 4 | 6 | 1 |
| **Offspring 1** | 7 | 2 | 3 | 4 | 6 | 0 | 4 | 1 | 5 | 9 |
| **Offspring 2** | 0 | 8 | 5 | 3 | 6 | 9 | 2 | 1 | 7 | 8 |

The sequence swap was got from the generated number in the system. The gen starts from 0 until 9. The system chooses gen no 1, 6, and 4. After the sequence swap must analysis to ensure the chromosome lack or excess of gen. From table 4 offspring 1 was lack of gen 8 and excess of gen 4. Offspring 2 lack of gen 4 and excess of gen 8. This condition should not happen, so the offspring will change. Below are the new offspring:

| Chromosome 1 | 7 | 8 | 3 | 4 | 6 | 0 | 2 | 1 | 5 | 9 |
| Chromosome 2 | 0 | 2 | 5 | 3 | 6 | 9 | 4 | 1 | 7 | 8 |
| **Sequence Swap** | 4 | 6 | 1 |
| **Offspring 1** | 7 | 2 | 3 | 8 | 6 | 0 | 4 | 1 | 5 | 9 |
| **Offspring 2** | 0 | 8 | 5 | 3 | 6 | 9 | 2 | 1 | 7 | 4 |

Table 5 show the final result of the crossover process, offspring 1 and offspring 2 will be the new generation. The new solution has two possibilities, the new solution can give a smaller tardiness than the old generation or can give a bigger tardiness than the old generation.

3.3.5 Mutation
Mutation process only need 1 chromosome to be an input to mutation process. Same with crossover, first must generate random number from 0 - 1. Below is the example of mutation probability:

| Chromosome 1 | 0,10676331 |
| Chromosome 2 | 0,003671227 |
| Chromosome 3 | 0,291121463 |
| Chromosome 4 | 0,402889211 |

Based on table 6, there is one chromosome have a smaller value than the parameter of the mutation probability. Chromosome 2 will be selected to mutation process. Below is the example of mutation process:
Table 7. Example of Mutation Process

| Chromosome 2 | 0 | 8 | 5 | 3 | 6 | 9 | 2 | 1 | 7 | 4 |
|-------------|---|---|---|---|---|---|---|---|---|---|
| **Sequence Swap** | 2 | 6 |
| Chromosome 2 | 0 | 8 | 9 | 3 | 6 | 5 | 2 | 1 | 7 | 4 |

Based on table 7, the system will generate from 0 to 9 and the selected number is 2 and 6. Gen 2 will replace by gen 6 and gen 6 will replace by gen 2. After the sequence swap, the chromosome 2 with new gen structure will be the new chromosome.

### 3.3.6 Survivor Selection
After the process of mutation new generation will be build by the process of mutation. After that the next process is to choose the survivor selection to find the best chromosome to be the solution of the problem.

### 4. Conclusion

#### 4.1 Comparison Result

The process of scheduling in this research using python language to make the program. Below is the comparison result the existing scheduling with scheduling using genetic algorithm in every machine:

| Machine | I1 | I2 | I3 | I4 | I5 | I6 | I7 | II1 | II2 | II3 | II4 | II5 | II6 | II7 | III1 | III2 | III3 | III4 | III5 |
|---------|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| **Tardiness (Hours)** | 304 | 216 | 280 | 464 | 480 | 312 | 512 | 600 | 296 | 392 | 504 | 288 | 432 | 264 | 592 | 624 | 400 | 600 | 574 |
| **Tardy Job** | 5 | 6 | 7 | 8 | 9 | 8 | 10 | 7 | 7 | 5 | 5 | 9 | 5 | 6 | 8 | 5 | 5 | 6 |
| **Tardiness (Hours)** | 118 | 103 | 31 | 37 | 4 | 125 | 27 | 0 | 77 | 25 | 37 | 0 | 0 | 45 | 26 | 31 | 0 | 38 |
| **Tardy Job** | 5 | 5 | 1 | 1 | 1 | 3 | 3 | 1 | 3 | 1 | 1 | 0 | 0 | 2 | 2 | 1 | 0 | 2 |

Based on table 8, genetic algorithm method in scheduling can reduce the total tardiness and total tardy jobs in every machine. GA has advantages when compared with exact algorithm and EDD Rule. Using exact method need a large time to solve problem with large size of jobs. Another constraint that sure GA is better than the exact method is when solve the problem using exact method with same python language only can solve maximum 10 jobs. In this case, if GA compared with EDD rule, using GA give a better solution in tardiness minimization. Using GA give a same loading in every machine so every machine has same jobs to processed. Using EDD give different loading of jobs in every machine. Some machine has more amount of jobs to be processed with the other machine. In lathe machine type I GA can minimize the total tardiness from 2568 hours to 443,85 hours or can reduce the tardiness of 82,71%. The total tardy job in existing scheduling are 51 jobs and with GA total tardy jobs are 19. In lathe machine type II, GA can minimize the total tardiness from 2776 hours to 139,314 hours or can reduce the tardiness of 94,98%. Total tardy job in lathe machine type II from 48 jobs reduce to be 6 jobs. In turret machine the existing tardiness are 2790 hours and with proposed GA the tardiness change to be 140,79 hours. Total tardy job in turret machine using existing schedule are 30 jobs and using GA reduce to be 7 jobs. Overall scheduling using the proposed method can reduce the tardiness from 8134 hours to be 723,97 hours or can reduce the tardiness of 91,10%. Total tardy job from 129 job reduce to be 32 jobs. Based on that, the proposed method can resolve the problem to minimization tardiness on parallel machine scheduling.
5. References:

[1] Pinedo ML 1984 Scheduling: Theory, Algorithms, and Systems (5th ed. Vol. 18) Journal of clinical orthodontics (JCO. New York, NY: Springer Cham Heidelberg New York Dordrecht London) 665 p

[2] Çiçekli UG 2016 A Genetic Algorithm To Minimize Makespan and Number of Tardy Jobs In Parallel Machine Scheduling Problems. Bilişim Teknol Derg. 9(2) 147–51

[3] Mokotof E 2001 Parallel Machine Scheduling Problems: A Survey. Asia-Pacific J Oper Res. 18 193–42 p

[4] Drobouchevitch IG, Sidney JB 2012 Minimization of earliness, tardiness and due date penalties on uniform parallel machines with identical jobs. Comput Oper Res [Internet]. 39(9) 1919–26

[5] Du J, Leung JY-T 1990 Minimizing Total Tardiness on One Machine is NP-Hard. Math Oper Res. 15(3) 483–495

[6] Eren T, Güner E 2007 Minimizing total tardiness in a scheduling problem with a learning effect. Appl Math Model. 31(7) 1351–61

[7] Sule DR 2008 Production Planning and Industrial Scheduling Examples, Case Studies and Applications Second Edi (New York, NY: CRC Press) 527 p

[8] Ginting, R 2009 Penjadwalan Mesin (Yogyakarta: Graha Ilmu) 270 p

[9] Bedworth, D. D., & Bailey, J. E 1987 Integrated Production Control Systems (Canada: Library of Congress Cataloging in Publication Data) 477 p

[10] Cheng, R., Gen, M., & Tozawa, T. 1995 Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. Computers and Industrial Engineering. 29(1–4) 513–517

[11] Michalewicz Z 1992 Genetic Algorithms + Data Structures = Evolution Programs First Edit (Charlote: Department of Computer Science University of North Carolina) 383 p

[12] Rivera G, Cisneros L, Sánchez-Solís P, Rangel-Valdez N, Rodas-Osollo J 2020 Genetic algorithm for scheduling optimization considering heterogeneous containers: A real-world case study. 9(1) 1–16

[13] Chaudhry, I. A., & Elbadawi, I. A. Q 2017 Minimisation of total tardiness for identical parallel machine scheduling using genetic algorithm. Sadhana - Academy Proceedings in Engineering Sciences. 42(1) 11–21

[14] Min, L., & Cheng, W 2006 Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems. Robotics and Computer-Integrated Manufacturing. 22(4) 279–287

[15] Demiral, T., Ozkir, V., Demiral, N., & Tasdelen, B 2012 A genetic algorithm approach for minimizing total tardiness in single machine scheduling. International Journal of Industrial Engineering and Management. 3(3) 163–171

[16] Sivapragasam, S., & Suppiah, Y 2017. Minimizing total weighted tardiness in identical parallel machine with sequence dependent setup time using genetic algorithm. Journal of Telecommunication, Electronic and Computer Engineering. 9(1–4) 89–93

[17] Liu, C 2013 A hybrid genetic algorithm to minimize total tardiness for unrelated parallel machine scheduling with precedence constraints. Mathematical Problems in Engineering. 2013 1–12

[18] M. O. Adamu 2012 Metaheuristics for scheduling on parallel machine to minimize weighted number of early and tardy jobs. Int J Phys Sci. 7(10) 1641–52

[19] Lingaraj, H 2019 A Study on Genetic Algorithms and its Applications. International Jornal Of Computer Sciences and Engineering. 4(10) 139–143

[20] Gen M, Cheng R, Lin L 2014 Network Models and Optimization First Edit Roy R, editor (British Library Cataloguing. London: British Library Cataloguing in Publication Data) 522 p

[21] Sivanandam S., Deepa S 2013 Introduction to Genetic Algorithms. First Edit. Vol. 53, Journal of Chemical Information and Modeling 1689–1699 p

[22] Wright AH 1991 Genetic Algorithms for Real Parameter Optimization. Found Genet Algorithms. 1 205–18