Jurnal Sistem Teknik Industri
Journal homepage: https://talenta.usu.ac.id/jsti

Parallel Scheduling using Genetic Algorithm and Knowledge Based Approach

Mentari Oktaria Gurusinga*[1], Rosnani Ginting[1], Sukaria Sinulingga[1]
[1]*Magister of Industrial Engineering, Faculty of Engineering, Universitas Sumatera Utara, Medan, 20155, Indonesia*
*Corresponding Author: mentarigurusingaa@gmail.com

ARTICLE INFO

Article history:
Received 8 December 2024
Revised 2 January 2025
Accepted 14 January 2025
Available online 29 April 2025

E-ISSN: 2527-9408
P-ISSN: 1411-5247

How to cite:
Gurusinga, M. O., Ginting, R., & Sinulingga, S., (2025). Parallel Scheduling using Genetic Algorithm and Knowledge Based Approach. *Jurnal Sistem Teknik Industri*, 27(2), 127-136.

http://doi.org/10.32734/register.v27i1.idarticle

ABSTRACT

Production scheduling are very important considering the complexity of the production system. This study aims to solve parallel machine scheduling to get the best job sequence and minimize lateness. Genetic algorithm is optimization algorithms by implementing evolution process and eliminating bad solutions. Knowledge based approach (KBA) solve problems by creating a computing system to imitates human intelligent behavior. Genetic algorithm and KBA are combined with the earliest due date (EDD) rule to produce an inference engine to build more adaptive population initialization. The results of the proposed scheduling show that the rules successfully guide the search process more adaptively. The genetic operation increasing the fitness value when the job is overload or underload. When the job is underload fitness increases by 3.56%, there is no lateness and load capacity ratio (LCR) increase by 4.67%. When the overload fitness increases by 1%, lateness decreases by 4.57%, and LCR decreases by 7.56%. The increase of fitness value shows better results of the proposed job sequence with minimum lateness. The implementation of integration genetic algorithms and KBA using VB.Net language requires a reasonable computing time, which is an average of 32 seconds when running.
Keyword: Scheduling, Genetic Algorithm, Makespan, Lateness, Flow Shop

ABSTRAK

Penjadwalan produksi sangat penting mengingat kompleksitas sistem produksi. Penelitian ini bertujuan menangani penjadwalan mesin paralel untuk mendapatkan urutan job dan meminimalkan keterlambatan. Algoritma genetika salah satu algoritma optimasi dengan menerapkan proses evolusi dan proses eliminasi solusi buruk. *Knowledge based approach* (KBA) membantu memecahkan masalah dengan penciptaan sistem komputasi meniru perilaku cerdas manusia. Algoritma genetika dan KBA dikombinasikan pada aturan earliest due date (EDD) menghasilkan algoritma pemilihan mesih sehingga membangun inisialisasi populasi awal lebih adaptif. Hasil usulan penjadwalan menunjukkan kombinasi aturan berhasil menuntun proses pencarian lebih adaptif. Proses operasi genetika meningkatkan nilai fitness saat job overload maupun underload. Saat *underload* fitness meningkat 3.56%, tidak terjadi keterlambatan dan utilisasi meningkat 4.67%. Saat *overload fitness* meningkat 1% keterlambatan menurun 4.57%, dan load capacity ratio (LCR) menurun 7.56%. Peningkatan nilai fitness menunjukkan hasil urutan job usulan lebih baik dengan nilai keterlambatan minimum. Implementasi integrasi algoritma genetika dan KBA dengan bahasa pemrograman VB.Net membutuhkan waktu komputasi wajar yaitu rata-rata 32 detik saat running.
Keyword: Scheduling, Genetic Algorithm, Makespan, Lateness, Flow Shop

1. Introduction

The development of the industrial era along with the increasing food and beverage, fashion, pharmaceutical, cosmetic, personal care and household industries has driven the growth of the sustainable packaging industry, one of which is the demand for corrugated box [1]. Corrugated box is lightweight and strong packaging usually used in transportation to protect products. This packaging is chosen as an alternative because made by environmentally friendly cellulose [2]. The growth in demand has attracted companies to optimize and improve performance in order to face competition.

Business competition shifts the focus of packaging production from make to stock become make to order to fulfill customer needs such as personal packaging designs. Production planning and operational scheduling are very important considering the complexity of the packaging production system such as the existence of just-in-time delivery rules, uncertain order arrivals, varying production volumes to optimizing utilities and just-in-time delivery [3]. Production scheduling is a technical decision-making process in the manufacturing industry related to the allocation of resources to complete tasks within a certain period of time in order to optimize one or more objectives [4].

Scheduling is part of operational planning that involves determining priority of orders that are truly ready to be worked on each workstation when the implementation period or schedule has arrived. Operation scheduling aims to provide the best results such as achieving timely completion (meeting due date), minimum lead time, minimum setup time and increasing workstation utilization. In practice, it is very difficult to find priorities that truly meet these criteria so that only one most important criterion is used as a basis for determining other priorities [5]. The combination of the criteria is a challenge in increasing efficiency, customer satisfaction and productivity.

Scheduling parallel machine flow shops with priority rules and stochastic job arrivals cannot be solved with classical parallel scheduling model [6]. In parallel flow shop scheduling, the sequence of jobs must be processed sequentially in set of machines without changing their process. Jobs must be processed and cannot be started at another stage if they have not been completed in the previous stage. The problem of parallel scheduling is difficult to solve even with one criteria, considering that more than that makes parallel scheduling more difficult to solve [7].

According to Özdöl [8] scheduling tends to be solved with optimization algorithms because some scheduling problems are included into difficult-to-solve. Genetic algorithms is the algorithms that have succeeded in finding solutions to various optimization problems because of ability to produce solutions by applying evolution process, eliminating bad solutions from first generation to next generation. Genetic algorithms widely used to solve various optimization problems because the steps are very simple and not need to find derivatives of functions to achieve global optimum values [4].

Genetic algorithms include five main phases namely population initialization, fitness function, crossover, mutation and selection [9]. Several previous studies have used genetic algorithms to solve scheduling problems, one of which can minimize total lateness and schedule disruptions due to urgent work [10]. This algorithm able to minimize makespan values effectively and efficiently because of the short time conversion in providing solutions through changes in sequence positions in chromosomes [11], [12], [13] simplifying the environmental structure and can be adjusted to the company's needs for product variations in terms of very large quantities and types [14], [15], [16]. Rolf et al. stated that the results of the solution will be more optimal if they can pay attention to shipping rules [17], because this is needed in the real practice of an industry. In addition to considering makespan, genetic algorithms will be better if they can solve multi-objective scheduling with several constraints [18], [19] such as considering due dates, scenarios without waiting time and considering production delay penalties [20]. Rashidi et al., solved multi objective parallel machine scheduling, namely minimizing delay and makespan simultaneously by converting them into single objective [21].

Knowledge based approach is an approach to collecting and utilizing knowledge data. The combination of a knowledge-based approach with an algorithm produces adaptive scheduling. The scheduling method can utilize the knowledge base by comparing current situation with knowledge base, thereby assisting scheduler in making decisions [22]. Pan et al., used a knowledge-based approach in a combination of the initial heuristic algorithm with modern heuristics as a knowledge base to reduce lateness [23]. The knowledge-based approach improves convergence and diversity of combinatorial parallel machine scheduling solutions [24] and facilitates adaptive scheduling decisions, thereby improving scheduling performance [25]. This study aims to design a proposed method combining genetic algorithm and knowledge-based approach to find job sequences by optimizing the load capacity ratio between machines to reduce lateness. The scenario applies the earliest due date rule with combination of knowledge-based approach as initialization of genetic algorithm. The balance of load capacity ratio increasing utilization of machine capacity in achieving company goals by reducing lateness.

2. Method

This study utilizes knowledge-based approach and genetic algorithm to solve parallel machines scheduling problem in corrugated box manufacturing industry. The stage of this study that is problem definition, knowledge-based approach and genetic algorithm.

2.1. Problem Definition

The parallel machine scheduling is the problem of assigning n jobs $j_1, j_2,, . . ., j_n$. Job j_i has a due date for priority customers $d_i \leq d_i + 1$ and for non-priority customers $d_i \leq d_i + 6$. There are k unrelated parallel machines $m_1, m_2,\ldots ,m_k$. with different specifications. Each machine operates with a machine speed depending the number of orders. The processing time machine is divided into R ranges. Jobs with a large number of orders result in the machine being able to operate at high speed, thereby saving processing time, where v_l is the speed index and r_o is the order range. The machine speed cannot change during job processing, p_i indicates the processing time of job j_i on machine m_k at speed v_l in the order rangerr_o. Setup time is the time to set the mold on the machine. One color uses a rubber mold so that setup time depends on number of colors in the job. The selection of machines for job considers to job criteria such as color, box length, width, height, type of knife, flute and processing flow. Each machine has a minimum and maximum size product so the job must be checked first to determine the machine assignment. Each machine can process at most one operation at a time, operations cannot be interrupted and all machines are available at all times. The scheduling problem in corrugated box manufacturing is illustrated in Figure 1.

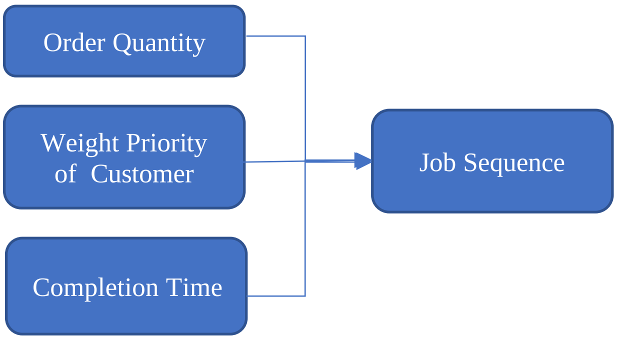


Figure 1. Scheduling illustration in corrugated box manufacturing industry

The problem is divided into three sub problems: 1) determining due date, 2) determining job completion time on machine assignment and 3) scheduling. The three sub problems are closely related. Determining completion time depends on machine assignment and order quantity, scheduling depends on the job selection so it is important to consider the job due date.

2.2. Knowledge Based Approach

Knowledge based approach related to computer programs with decision-making capabilities to solve a problem related to creation of a computing system that imitates intelligent behavior of human expertise. This system has characteristics of adaptive control, handling, and reuse of knowledge by performing actions such as perception, interpretation, reasoning, learning, communication and decision making in finding solutions to problems. Expert knowledge is encoded into facts, rules, heuristics, and procedures through databases, knowledge bases and inference engines. The inference engine acts as a controlling environment and interacts with users, understanding the knowledge base to formulate conclusions and provide expert advice in solving problems [26]. Integration of genetic algorithm and knowledge-based approach reduces the searching solution's time by providing more structured guidance. Knowledge based approach generates initial solution, guides mutation, improves solution in genetic algorithm [23]. Knowledge based approach improves convergence and diversity of combinatorial parallel machine scheduling solutions [24] and facilitates adaptive scheduling decision thus improving scheduling performance [25]. The basic structure of knowledge based scheduling system is seen in Figure 2.

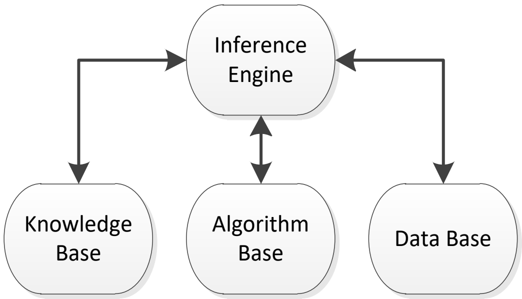


Figure 2. Knowledge Based System Scheduling Structure

The knowledge-based approach aims to collect information as knowledge base data in database. This stage begins with identification of production process, machine criteria and job specifications to determine the inference engine based on process time and machine capacity [27]. The limitations of machine selection, machine capacity and customer priority are saved in a table of knowledge data. The table will be used as a reference of knowledge representation. The knowledge representation stage extracts information, determines job execution deadlines, creates inference engines, calculates utilization and lateness by tracking and matching rules with facts in database of job collection [28]. The information of machine selection is combined with earliest due date rule by first defining due date according to customer priority. The stages in knowledge-based approach are shown in Table 1.

Table 1. Knowledge-Based Algorithm

No	Algorithm
1	Input processing time of each machine based on range in database
2	Input setup time based on number of colors in database
3	Input customer priority in database
4	Input machine specification in database (color, length, width, height product, type of knife, flute, flow process)
5	Match job with machine specification in database
6	Match order quantity with order range in database
7	Calculate processing time = processing time x number of orders
8	Match job color quantity and calculate setup time
9	Calculate completion time: processing time + setup time
10	Match customer name in database, determine due date job. Customer priority due date tolerated one day and customer non priority tolerated six days
11	Generate alternative job allocation options on each machine compatible with job completion time and due date information

2.3. Genetic Algorithm

Independently, scheduling is the process of organizing, controlling, and optimizing work and workloads in a production process or manufacturing process. Optimization is process of solving a particular problem so that it is in the most advantageous condition from a point of view [29]. The difficulty in optimization process is determining optimum value obtained. The value in question is determining global optimum value or simply local optimum. The local optimum is optimal value achieved within a certain range of values. While the global optimum is optimal value of all members of population [30]. Flow shop is a production system where all jobs follow same sequence through a number of machines or workstations. In a parallel flow shop production system, there is more than one parallel machine at each stage to process same job and allows flexibility in assigning jobs to machines, thus minimizing makespan, lateness, and load capacity ratio [31].

Parallel flow shop scheduling with simple dispatching rule and classical heuristics produces local optimum values, so it needs to be combined with an optimization algorithm to obtain global optimum value. Genetic algorithm is one of the evolutionary optimization algorithms that is often used in solving parallel machine flow shop scheduling. Genetic algorithms effectively obtain job execution sequences to minimize lateness and completion times [6], [7], [21]. The basic structure of genetic algorithm shown at Figure 3.

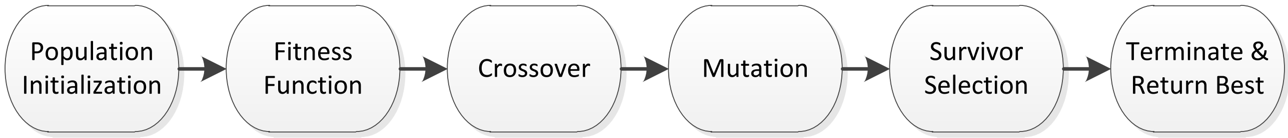


Figure 3. Basic Structure of Genetic Algorithm

Genes represent production command jobs on a machine. The job allocation for each machine form of an array or list of jobs representing parallel machines with an initial population size of 100 chromosomes. The job ID and job index indicate the order in which tasks are executed. Each gene arranged into a chromosome to produce a sequence and allocation jobs on the machine using earliest due date rule with illustrations such as Figure 4.

Machines	Job List
M 1	J1, J2, .. Jn
M 2	J3, J4, .. Jn
M 3	J5, .. Jn
M 4	J6, J7, .. Jn
M 5	J8, J9, .. Jn
M 6	J10, .. Jn
M 7	J11, J12 .. Jn
M 8	J13, J14, .. Jn
M 9	J15 .. Jn

Figure 4. Chromosome

There are nine genes, where each gene is a sequence of jobs that can be processed on each machine. Each job is allocated exactly once to one machine. All jobs are allocated evenly to all machines. The job sequence will be selected for its fitness value through the fitness function. The fitness function aims to allocate job to parallel machines, select the appropriate machine to allocate each job, and determine the sequences of job on each machine to minimize two objectives simultaneously that to reducing lateness [24] and increasing load capacity ratio [32] with the function:

$$\max f_1 = \frac{1}{1+\frac{\sum_{i=1}^{n}\left|1-\left(\frac{C_{maks}}{C_{total}}\right)\right|}{n}} \quad (1)$$

$$\max f_2 = \frac{1}{1+\sum_{i=1}^{n} T_i} \quad (2)$$

Where, T_i = max $\{C_i - D_i, 0\}$ show lateness job j_i. C_i shows completion time of job j_i, f_1 shows total load capacity ratio, dan f_2 is total lateness. If job j_i is a job with a range of orders r_o allocated to the machine m_k with speed v_l, z_{ikl}= 1, otherwise 0. Fitness evaluation of determining how optimally chromosomes meet the objectives of the problem. The higher fitness value, more optimal chromosome meets the objective function. Fitness values are evaluated when job is underloaded and overloaded to ensure value representative of objectives in each job arrival condition. Ranking fitness values prevents solutions from getting stuck in local optima conditions. The rank-based selection method sorts population based on highest value [33]. Parent chromosomes are selected before crossover process using probability 0.8.

One point crossover exchanging genetic of two chromosomes parts at one point that is determined or randomly selected to produce offspring. This process aims to combine characteristics of both parents and create variation in a new population in form of offspring chromosomes. In this study, one chromosome exchange point was taken randomly on a compatible machine. The offspring fitness value obtained increased compared to the parent fitness. Offspring was inserted into the population to replace parent and then re-selected for mutation process [34].

Mutation introduces genetic variation into a new population by randomly changing position of genes in chromosome. Mutation prevents stagnation in local solutions and maintains population diversity so that exploration of solution space becomes wider. The mutation probability of 0.05 keeps changes to a minimum and avoids drastic changes that can damage chromosomes. Swap mutation technique changes relative positions of two jobs in a chromosome randomly, maintaining number of jobs and their sequences. The mutated chromosome inserted into the population to replace old chromosome [33].

The chromosome with the highest fitness value from population initialization or genetic operations is selected to ensure that chromosome with the best performance remains in next generation [9]. The genetic algorithm stage is shown in Table 2.

Table 2. Genetic Algorithm

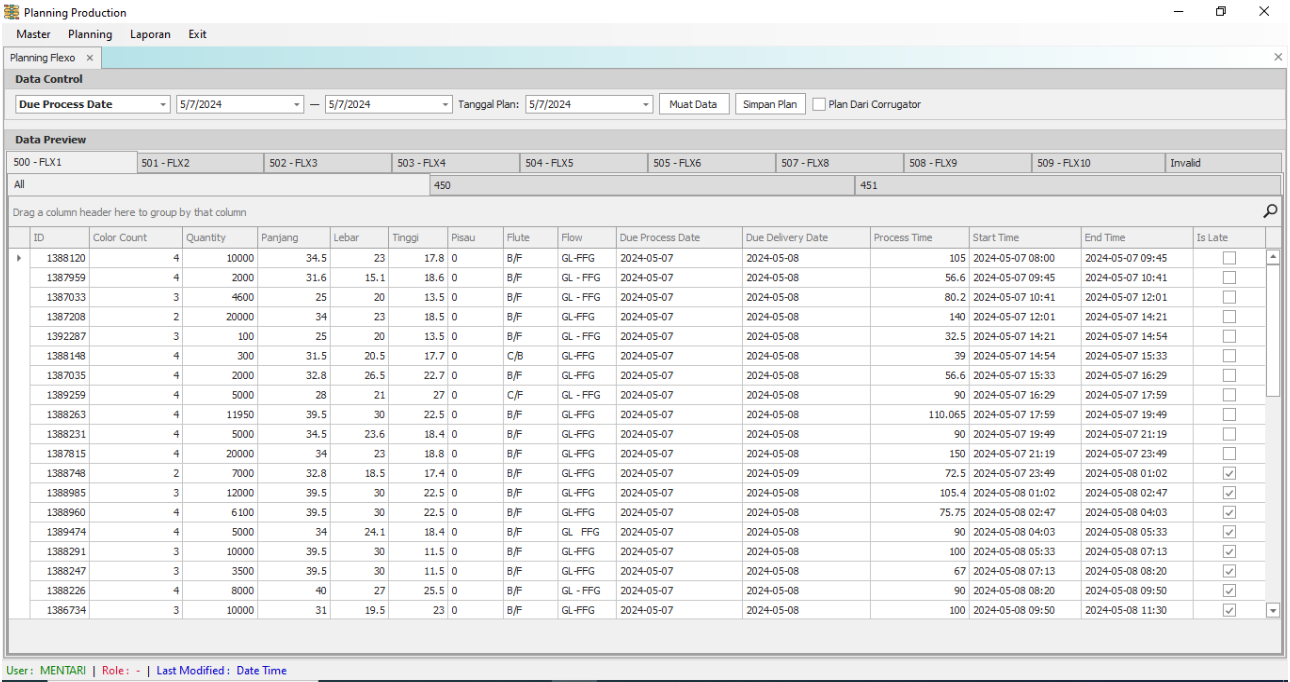
No	Algorithm
1	Select job using rules earliest due date
2	Decide minimum work load off all machine, allocate job to compatible machine randomly
3	Create a sequence of parallel machine jobs with 100 alternative solutions as initial population
4	Calculate total completion time of each machine.
5	Calculate load capacity ratio = |1- (completion time / machine capacity) |
6	Calculate average load capacity ratio chromosome = 1/ (1+|1- (completion time / machine capacity) |)
7	Allocate each job to a machine with a shift start time of 08.00 a.m then calculate the number of late jobs. A job is late if it passes the cut off time at 12.00 p.m.
8	Calculate total lateness of chromosome = 1 / (1 + number of late jobs).
9	Calculate fitness value chromosome = 0.8 x load capacity ratio + 0.2 x lateness.
10	Evaluate fitness value with rank-based selection.
11	Generate random numbers to select parent chromosome.
12	If random / fitness value < crossover probability (0.8), chromosome is selected as parent while other chromosome remains in population
13	Crossover parent chromosome with one-point crossover becomes offspring chromosome.
14	Calculate fitness value of the offspring chromosome and reinsert it into population
15	Generate random numbers to select chromosomes for mutation. Each job with a random value < mutation probability (0.05) will go through the mutation stage while other chromosomes remain in population.
16	Mutate selected chromosome using swap mutation technique.
17	Calculate fitness chromosome and re-insert chromosome into population.
18	Select chromosome with the highest fitness value as the best chromosome.

3. Result and Discussion

Pan et al., stated that development of a knowledge-based optimization algorithm is able to minimize total energy consumption and total delay by integrating plant and machine allocation. Knowledge of plant allocation, machine allocation, and scheduling simplifies the complexity of decision making. The knowledge-based approach ensures that the genetic algorithm utilizes an understanding of the problem structure to improve solution search, avoid unproductive searches and focus more on solutions [23].

The integration of knowledge-based approach and genetic algorithm in visual studio aims to create an automatic, intelligent, and effective production scheduling system so that it can increase the productivity and competitiveness of the company. The application of the integration of genetic algorithm and knowledge-based approach creates an adaptive scheduling system as a simulation program to help determine the sequence of jobs to reduce production lateness. The integration model is implemented in the form of a desktop program / application with the vb.net programming language and the Integrated Development Environment (IDE) Visual Studio software [35]. Genetic algorithms can help avoid job accumulation at critical points in the production process, while knowledge-based approaches provide rules to consider human expertise and experience in handling special scenarios. The combination of the two produces a system to find the optimal schedule by reducing production time, increasing resource utilization and making it easier to adjust the schedule according to real-time conditions [25].

The development of an inference engine is tasked with processing and interpreting knowledge, producing scheduling decisions or recommendations. The inference engine takes information from the knowledge base, applies scheduling rules/logic, and provides optimal solutions. The user interface is created by considering the need to make it easier for users to determine production process input. The user interface consists of a form display to enter data that affects scheduling such as machine specification limitations, machine capacity, process time, setup time, and due date [26]. Computation time produces schedules and solutions according to the time limit in production practice, which is an average of 32 seconds. The user interface are show in Figure 5.



ID	Color Count	Quantity	Panjang	Lebar	Tinggi	Pisau	Flute	Flow	Due Process Date	Due Delivery Date	Process Time	Start Time	End Time	Is Late
1388120	4	10000	34.5	23	17.8	0	B/F	GL-FFG	2024-05-07	2024-05-08	105	2024-05-07 08:00	2024-05-07 09:45	☐
1387959	4	2000	31.6	15.1	18.6	0	B/F	GL - FFG	2024-05-07	2024-05-08	56.6	2024-05-07 09:45	2024-05-07 10:41	☐
1387033	3	4600	25	20	13.5	0	B/F	GL - FFG	2024-05-07	2024-05-08	80.2	2024-05-07 10:41	2024-05-07 12:01	☐
1387208	2	20000	34	23	18.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	140	2024-05-07 12:01	2024-05-07 14:21	☐
1392287	3	100	25	20	13.5	0	B/F	GL - FFG	2024-05-07	2024-05-08	32.5	2024-05-07 14:21	2024-05-07 14:54	☐
1388148	4	300	31.5	20.5	17.7	0	C/B	GL-FFG	2024-05-07	2024-05-08	39	2024-05-07 14:54	2024-05-07 15:33	☐
1387035	4	2000	32.8	26.5	22.7	0	B/F	GL-FFG	2024-05-07	2024-05-08	56.6	2024-05-07 15:33	2024-05-07 16:29	☐
1389259	4	5000	28	21	27	0	C/F	GL - FFG	2024-05-07	2024-05-08	90	2024-05-07 16:29	2024-05-07 17:59	☐
1388263	4	11950	39.5	30	22.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	110.065	2024-05-07 17:59	2024-05-07 19:49	☐
1388231	4	5000	34.5	23.6	18.4	0	B/F	GL-FFG	2024-05-07	2024-05-08	90	2024-05-07 19:49	2024-05-07 21:19	☐
1387815	4	20000	34	23	18.8	0	B/F	GL-FFG	2024-05-07	2024-05-08	150	2024-05-07 21:19	2024-05-07 23:49	☐
1388748	2	7000	32.8	18.5	17.4	0	B/F	GL-FFG	2024-05-07	2024-05-09	72.5	2024-05-07 23:49	2024-05-08 01:02	☑
1388985	3	12000	39.5	30	22.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	105.4	2024-05-08 01:02	2024-05-08 02:47	☑
1388960	4	6100	39.5	30	22.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	75.75	2024-05-08 02:47	2024-05-08 04:03	☑
1389474	4	5000	34	24.1	18.4	0	B/F	GL FFG	2024-05-07	2024-05-08	90	2024-05-08 04:03	2024-05-08 05:33	☑
1388291	3	10000	39.5	30	11.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	100	2024-05-08 05:33	2024-05-08 07:13	☑
1388247	3	3500	39.5	30	11.5	0	B/F	GL-FFG	2024-05-07	2024-05-08	67	2024-05-08 07:13	2024-05-08 08:20	☑
1388226	4	8000	40	27	25.5	0	B/F	GL - FFG	2024-05-07	2024-05-08	90	2024-05-08 08:20	2024-05-08 09:50	☑
1386734	3	10000	31	19.5	23	0	B/F	GL-FFG	2024-05-07	2024-05-08	100	2024-05-08 09:50	2024-05-08 11:30	☑

Figure 5. User Interface

The system can quickly and efficiently re-schedule and maintain performance despite changes in capacity and processing time. Output data validation ensures that the schedule complies with business rules and does not violate production constraints. There is no conflict or inability to produce a job sequence as long as the input data is complete and correct. Scheduling system verification ensures that the system generates production schedules efficiently, meets business needs, and is adaptive to change. Verification checks the validation of machine selection to ensure that the system selects only valid values from the list of available machine specification choices and allocates the job only once and to one machine. Verification is done by manually comparing the program results.

Fitness value verification ensures that the results of the genetic algorithm truly reflect the quality of the chromosome solution in the population by ensuring that the fitness function produces valid and consistent values. The genetic operation process successfully increases the fitness value from initialization to the final process when the job is overloaded or underloaded. During job overload, the highest chromosome fitness value of the initial population increases by 3.56% in the final population. During job underload, the highest chromosome fitness value of the initial population increases by 3.56% in the final population. During job overload, the highest chromosome fitness value of the initial population increases by 1% in the final population. This implies that the genetic operator successfully improves the job sequence to be better. The increase in fitness value can be seen in Figure 6.

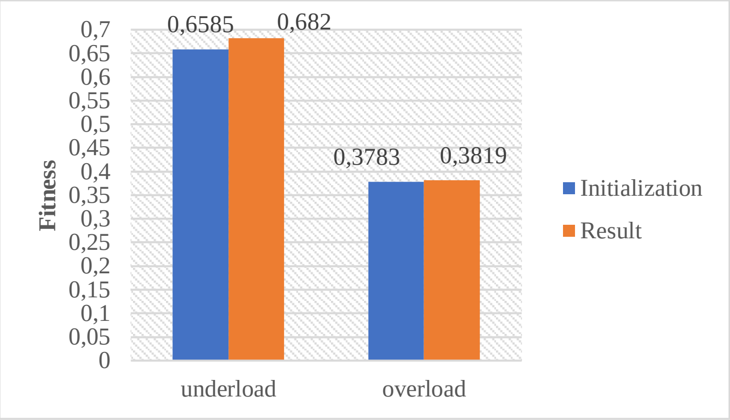


Figure 6. The Increase Fitness Value

The job sequence machine utility resulting from the algorithm increases the efficiency of job allocation on the machine when the job is overloaded or underloaded. When the job is underloaded, the machine utilization increases, while when the job is overloaded, the machine utilization decreases towards the optimum value. When the job is underloaded, the highest load capacity ratio value in initial population increases by 4.67% in the final population. When job is overloaded, highest chromosome load capacity ratio decreases by 7.56%. This implies that algorithm can optimize machine utilization through more even job allocation on machine. A comparison of load capacity ratio can be seen in Figure 7.

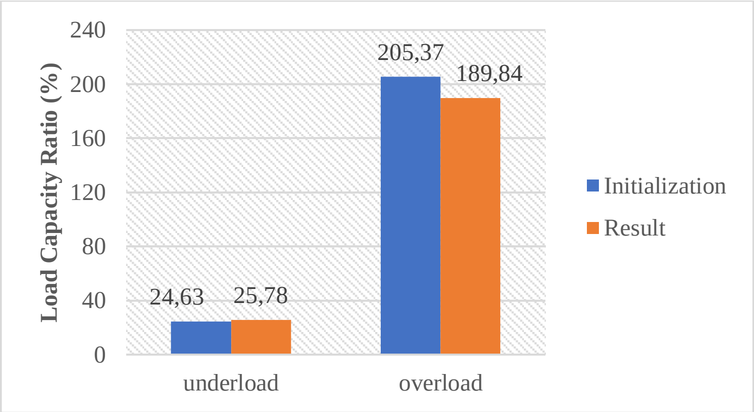


Figure 7. The Increase Machine Load Capacity Ratio

The proposed design reduces lateness during job overload. The comparison lateness of the job initialization sequence and the final result can be seen in Figure 8.

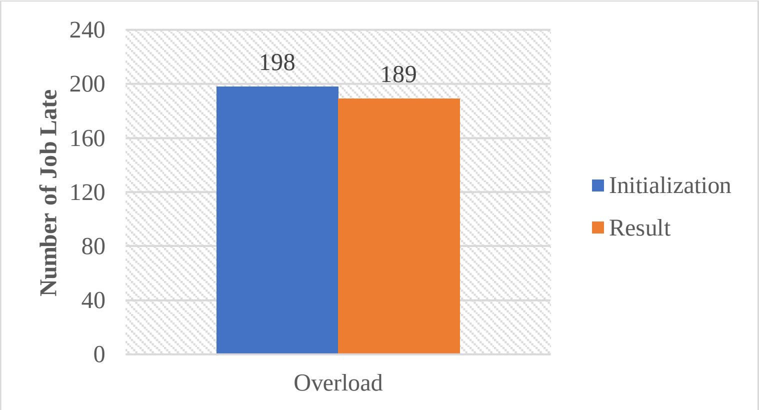


Figure 8. The Decrease of Lateness

Genetic algorithms and knowledge based approaches result more optimal job sequences where decreasing lateness and more precise and even machine selection. Genetic operators increase the fitness value from the initial population when job overloaded or underloaded. When job is underload fitness increases by 3.56%, there is no lateness and load capacity ratio increase by 4.67%. When the overload fitness increases by 1%, lateness decreases by 4.57%, and load capacity ratio decreases by 7.56%. The increase of fitness value shows better results of the proposed job sequence with minimum lateness. The results of the proposed job scheduling sequence can meet the company's objectives by reducing lateness and increasing fitness values.

4. Conclusion

The proposed scheduling system helps scheduler to make an effective plan by applying genetic algorithm and knowledge-based approach. The results of proposed scheduling show that earliest due date rule and knowledge-based approach successfully guide the search process to be more adaptive to changes. Genetic operators increase fitness value from the initial initialization to the final process when the job is overload or underload. Implementation of integration genetic algorithm and knowledge-based approach with the VB.Net programming language requires an average computing time of 32 seconds when running. This duration is more efficient compared to manual calculations. Overall, the proposed system produces a more optimal job execution sequence where there are no late because machine allocation is more balanced.

This study focuses on main objective to minimizing lateness. Further research can expand the scope to consider several objectives at once, such as energy efficiency, waste reduction, and customer satisfaction levels. This research is limited because it only uses a metaheuristic algorithm with a dispatching rule, namely a genetic algorithm with the earliest due date rule. Further research can compare several dispatching rules as an alternative to selecting the best job sequence or using other heuristic algorithms to compare better results.

Tabu search or swarm intelligence algorithms can continue to be developed and adapted to overcome new challenges in parallel machine scheduling.

References

[1] I. Jestratijevic and U. Vrabič-Brodnjak, "Sustainable and Innovative Packaging Solutions in the Fashion Industry: Global Report," *Sustain.*, vol. 14, no. 20, Oct. 2022, doi: 10.3390/su142013476.

[2] R. Fitas, H. J. Schaffrath, and S. Schabel, "A Review of Optimization for Corrugated Boards," Nov. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/su152115588.

[3] R. Ojstersek, M. Brezocnik, and B. Buchmeister, "Multi-objective optimization of production scheduling with evolutionary computation: A review," 2020, *Growing Science*. doi: 10.5267/j.ijiec.2020.1.003.

[4] R. Ginting, *Penjadwalan Mesin: Sistem, Algoritma, Pemecahan Masalah dan Penerapan*. Medan: USU Press, 2022.

[5] S. Sinulingga, *Perencanaan & Pengendalian Produksi*. Medan: USU Press, 2021.

[6] P. Perez-Gonzalez and J. M. Framinan, "A review and classification on distributed permutation flowshop scheduling problems," Jan. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.ejor.2023.02.001.

[7] A. Allahverdi, "A new heuristic for m-machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness," *Comput. Oper. Res.*, vol. 31, no. 2, pp. 157–180, Feb. 2004, doi: 10.1016/S0305-0548(02)00143-0.

[8] E. Özdöl, "A Genetic Algorithm for a Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times," Dokuz Eylül University, Ontario, 2011.

[9] P. Kianpour, D. Gupta, K. Krishnan, and B. Gopalakrishnan, "DOE-based Enhanced Genetic Algorithm for Unrelated Parallel Machine Scheduling to Minimize Earliness and Tardiness Costs," *J. Optim. Ind. Eng.*, vol. 16, no. 2, pp. 99–114, Jun. 2023, doi: 10.22094/JOIE.2023.1967925.1992.

[10] F. Abderrabi *et al.*, "Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Time and Job Splitting: Hospital Catering Case Study," *Appl. Sci.*, vol. 11, no. 4, pp. 1–27, Feb. 2021, doi: 10.3390/app11041504.

[11] D. Rooyani and F. M. Defersha, "An efficient two-stage genetic algorithm for flexible job-shop scheduling," in *IFAC-PapersOnLine*, Elsevier B.V., Sep. 2019, pp. 2519–2524. doi: 10.1016/j.ifacol.2019.11.585.

[12] F. M. Defersha and D. Rooyani, "An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time," *Comput. Ind. Eng.*, vol. 147, Sep. 2020, doi: 10.1016/j.cie.2020.106605.

[13] L. Meng *et al.*, "Constraint programing for solving four complex flexible shop scheduling problems," *IET Collab. Intell. Manuf.*, vol. 3, no. 2, pp. 147–160, 2021, doi: 10.1049/cim2.12005.

[14] N. Xie and N. Chen, "Flexible job shop scheduling problem with interval grey processing time," *Appl. Soft Comput. J.*, vol. 70, 2018, doi: 10.1016/j.asoc.2018.06.004.
[15] L. Cheng, Q. Tang, L. Zhang, and C. Yu, "Scheduling flexible manufacturing cell with no-idle flow-lines and job-shop via Q-learning-based genetic algorithm," *Comput. Ind. Eng.*, vol. 169, Jul. 2022, doi: 10.1016/j.cie.2022.108293.
[16] J. Fan, C. Zhang, Q. Liu, W. Shen, and L. Gao, "An improved genetic algorithm for flexible job shop scheduling problem considering reconfigurable machine tools with limited auxiliary modules," *J. Manuf. Syst.*, vol. 62, pp. 650–667, Jan. 2022, doi: 10.1016/j.jmsy.2022.01.014.
[17] B. Rolf, T. Reggelin, A. Nahhas, S. Lang, and M. Müller, "Assigning dispatching rules using a genetic algorithm to solve a hybrid flow shop scheduling problem," *Procedia Manuf.*, vol. 42, no. 2019, pp. 442–449, 2020, doi: 10.1016/j.promfg.2020.02.051.
[18] S. Yuan, T. Li, and B. Wang, "A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times," *Expert Syst. Appl.*, vol. 152, p. 113360, 2020, doi: 10.1016/j.eswa.2020.113360.
[19] M. S. Umam, M. Mustafid, and S. Suryono, "A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7459–7467, 2021, doi: 10.1016/j.jksuci.2021.08.025.
[20] P. D. Kusuma, "Multi-objective Batch Scheduling in Collaborative Multi-product Flow Shop System by using Non-dominated Sorting Genetic Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 349–357, 2021, doi: 10.14569/IJACSA.2021.0120939.
[21] E. Rashidi, M. Jahandar, and M. Zandieh, "An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines," *Int. J. Adv. Manuf. Technol.*, vol. 49, no. 9–12, pp. 1129–1139, Aug. 2010, doi: 10.1007/s00170-009-2475-z.
[22] M. Aufenanger, N. Lipka, B. Klopper, and W. Dangelmaier, *A Knowledge-Based Giffler-Thompson Heuristic for Rescheduling Job Shops*. IEEE, 2009.
[23] Z. Pan, D. Lei, and L. Wang, "A Knowledge-Based Two-Population Optimization Algorithm for Distributed Energy-Efficient Parallel Machines Scheduling," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5051–5063, Jun. 2020, doi: 10.1109/TCYB.2020.3026571.
[24] H. Wang, R. Li, and W. Gong, "Minimizing tardiness and makespan for distributed heterogeneous unrelated parallel machine scheduling by knowledge and Pareto-based memetic algorithm," *Egypt. Informatics J.*, vol. 24, no. 3, Sep. 2023, doi: 10.1016/j.eij.2023.05.008.
[25] Z. Qin and Y. Lu, "Manufacturing Letters A Knowledge Graph-based knowledge representation for adaptive manufacturing control under mass personalization-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0) Peer-review under responsibility of the Scientific Committee of the NAMRI/SME," 2023, [Online]. Available: www.sciencedirect.com
[26] S. P. Leo Kumar, "Knowledge-based expert system in manufacturing planning: state-of-the-art review," 2019, *Taylor and Francis Ltd.* doi: 10.1080/00207543.2018.1424372.
[27] A. Kusiak, "KBSS: A Knowledge-Based System for Scheduling in Automated Manufacturing," 1990.
[28] H. Sirait, E. U. Simanihuruk, R. Tambun, A. Parbina Nusantara, and P. Siantar, "Elemen Dasar dan Kharakteristik Membangun Sistem Pakar," *J. BisantaraInformatika*, vol. 7, no. 1, 2023.
[29] Z. Zukhri, *Algoritma Genetika, Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi, 2014.
[30] T. Weise, *Global Optimization Algorithms – Theory and Application –*. 2009. Accessed: Sep. 10, 2024. [Online]. Available: http://www2.fiit.stuba.sk/~kvasnicka/Free books/Weisse_Global Optimization Algorithms - Theory and Application.pdf
[31] M. L. Pinedo, "Scheduling: Theory, Algorithms, and Systems," 2002.
[32] W. J. Stevenson, *Operations Management*, 13th ed. New York: McGraw-Hill Education, 2018.
[33] A. Syarif, *Algoritma Genetika, Teori dan Aplikasi*, Kedua. Bandar Lampung: Graha Ilmu, 2014.
[34] K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms," 2001.
[35] M. Tonggiroh, V. B. A. Pardosi, Basiroh, and F. Nugroho, *Rekayasa Perangkat Lunak*, 1st ed. Solok: PT Mafy Media Literasi Indonesia, 2024.