E-ISSN: 2664-8784
P-ISSN: 2664-8776
Impact Factor: RJIF 8.26
IJRE 2025; 7(2): 162-170

www.engineeringpaper.net
Received: 24-08-2025
Accepted: 28-09-2025

Pak MC
Institute of Robot Engineering, Kim Chaek University of Technology, Kyogu-60, Sungni Street, Pyongyang, Democratic, People's Republic of Korea

Jang KW
Institute of Robot Engineering, Kim Chaek University of Technology, Kyogu-60, Sungni Street, Pyongyang, Democratic, People's Republic of Korea

Correspondence
Pak MC
Institute of Robot Engineering, Kim Chaek University of Technology, Kyogu-60, Sungni Street, Pyongyang, Democratic, People's Republic of Korea

Optimization of job-shop scheduling minimizing makespan in FMS using genetic algorithm and FlexSim Software

Pak MC and Jang KW

DOI: https://www.doi.org/10.33545/26648776.2025.v7.i2b.136

Abstract
The job-shop scheduling problem is one of the combinatorial optimization problems, which is very difficult to find the optimal solution in its nonlinear nature and reasonable time. This paper selects the job-shop scheduling problem as a research object and constructs a mathematical model aimed at minimizing the makespan in flexible manufacturing system (FMS). The genetic algorithm is applied to the scheduling program to obtain the optimal solution in the job-shop scheduling problem. A new solution method is proposed to get the optimal scheduling result according to the selection of the mutation type, using a job-shop scheduling program based on genetic algorithm. The Flexsim based on 3D discrete event simulation is used to simulate and verify the job-shop scheduling results obtained by selection of mutation type. Finally, through analyzing the simulated results, it shows that the proposed method is efficient and reasonable for solving job-shop scheduling problem in FMS.

Keywords: Job-shop Scheduling Problem (JSSP), Genetic Algorithm (GA), FlexSim, Mutation, Makespan, Flexible Manufacturing System (FMS)

Introduction

Flexible manufacturing systems are a combination of automated material handling systems and computer-aided numerical control machines controlled by distributed computer-controlled systems [1]. Today, planning and managing production in FMS is a very complex task, so the scheduling problem is an important decision-making problem. Effective scheduling increases the efficiency of the system, so we have to design a suitable scheduling procedure to optimize the performance indices of the system. Flexible manufacturing system includes job-shop scheduling and flow-shop scheduling. Where, the job-shop scheduling problem (JSSP) is the most difficult combinatorial optimization problem due to its NP-hard problem. This problem is difficult to solve with exact algorithms within the available time. The optimization for job-shop scheduling problem includes makespan minimization, total cost minimization, and resource minimization. The present study focuses on minimizing makespan.

In recent years, many scholars introduce and put forward a variety of heuristic algorithm to solve job-shop scheduling problems. Among them, genetic algorithms have been widely used and developed for its good robustness and stability [2, 3]. Moreover, Genetic algorithm shows its good effectiveness for solving job-shop scheduling problems [4-6]. In the job-shop scheduling problem, a mathematical model aimed at minimizing the maximum makespan was constructed[7]. The authors designed a solution model based on genetic algorithms and verified its effectiveness using DELMIA/QUEST simulation software. Also, scholars have applied genetic algorithms and their based programs to minimize the makespan in job shop scheduling problems [8-9]. Nikhil [10] generated different schedules and analyzed performances with priority rules using LEKIN software. The results showed that the shortest processing time (SPT) has optimal scheduling values than other priority rules. Yadav [11] proposed two types of scheduling algorithms, namely priority scheduling (TMFR and STPT) and genetic algorithm, and compared the performance of the algorithms in MATLAB. The results showed that the genetic algorithm performed better in minimizing makespan.

Ali [12] proposed a job-shop scheduling problem with machine setup time and applied genetic algorithm with the aim of minimizing makesan. The distributed flexible job-shop scheduling problem (DFJSP) that minimizes the makespan has been solved using hybrid algorithm and variable neighborhood search (VNS) [13].

On the other hand, simulation methods have been widely used to analyze the performance characteristics of the system. Currently, several simulation software products have been developed for discrete event simulation, including scheduling optimal solutions. The discrete event simulation software FlexSim is 3D object-oriented simulation software that allows model creation, simulation, analysis and optimization for discrete event systems, including manufacturing processes and material handling. Recently, researchers have investigated simulation methods for scheduling optimization in manufacturing processes using FlexSim. Raju [14] used the shortest machining technique (SPT) for job-shop scheduling problem (JSSP) solution in FlexSim simulation environment. The main goal of the simulation is to optimize the total processing time of JSSP by reducing the idle time of the machine and increasing the utilization rate of the machines. Krynke [15] used FlexSim to solve the problems of personnel allocation to the production line, and Pontes [16] used Flexsim to simulate cost optimization and machining process optimization in manufacturing processes including flexible production cells. Gopinath [17] simulated an improved routing optimization in Flexsim, Kumar [18] applied Flexsim for performance measurement and analysis of flexible manufacturing systems, and KLIMENT [19] described the problems in creating a simulation model in Flexsim environment.

Most previous studies have applied various genetic algorithms to solve Job-Shop scheduling problems in manufacturing processes, including flexible manufacturing systems, and analyzed and verified the optimal solution using simulation techniques as FlexSim. In genetic algorithm, the natures of the next generation created by crossover are determined by mutation. Therefore, the nature of the offspring depends on the mutation method. Hence, the selection of the mutation type is very an important in solving the job- shop scheduling problem.

The aim of this paper is to minimize the makespan in FMS. A job-shop scheduling program based on genetic algorithm is used to obtain the optimal scheduling result. The results show that the choice of the mutation type in job-shop scheduling problem affects the minimization of the makespan. The simulation method by FlexSim is effective and reasonable to verify the optimized job-shop solution in manufacturing process systems including flexible manufacturing systems. This paper is organized as follows. Section 2 describes the related details of job-shop scheduling problem in FMS and constructs a mathematical model with minimization of makespan as an objective function. Section 3 explains the proposed genetic algorithm to solve the JSSP and proposes a new solution method to obtain the optimal scheduling results according to the mutation type using the genetic algorithm-based job-shop scheduling program. Section 4 describes the simulation method to simulate and evaluate the obtained optimal scheduling results, using FlexSim software. FlexSim creates a FMS model using the objects and libraries it provides. The optimal scheduling data are connected to each object of the model to simulate according to the schedule. Section 5 analyzes and discusses the results obtained by the proposed method. Finally, Section 6 summarises some conclusions and further works.

2 Descriptions and modeling for job-shop scheduling problem

2.1 Problem description

Job-shop scheduling is the order of jobs in a machine processing to ensure that all parts complete the processing in a reasonable total processing time. The scheduling objective is to minimize the maximum production flow time (makespan) by determining the machining sequence and machining time of workpieces at all machines in the system according to the machining process constraints. To describe the problem we give the following assumptions. The FMS work plan is as follows:

Assuming there are m different machines and n different workpieces,

- Each workpiece contains a set of processes consisting of several processes, and the order of the workpiece is predefined;
- The machining time of the process sequence in each workpiece is known and constant;
- All processes of each workpiece can be machined on the machine in the system;
- For each process of the workpiece, it can provide a set of alternative machines consisting of several machines;
- The order of processing of each workpiece in each machine is completely different;
- The order of processing on each machine for each workpiece is completely different.
- Let i = 1, 2, ... M represent different machines, and j = 1, 2, ... N represent different jobs.
- For FMS, Job-Shop scheduling must satisfy the following constraints:
- The workpiece cannot access the same machine twice;
- Any workpiece can be machined in the next process if machining is completed in the previous process;
- After one machining task is completed on the same machine, another machining task can be started;
- There is no precedence constraint on the order of different workpieces;
- The workpiece is machined simultaneously on only one machine, and machine is machined simultaneously on one workpiece;
- The gap between each workpiece and the adjacent process is zero;
- Once the process is started, you must continue it.

2.2 Mathematical modeling

Under the above assumptions, it is constructed the mathematical model of Job-Shop scheduling problem with the objective of optimizing the minimization of makespan as follows.

$$\min Z = \max_{1\le k\le m}\left\{\max_{1\le i\le n} C_{ik}\right\} \tag{1}$$

Where

$$c_{ik} - p_{ik} + M\left(1 - a_{ihk}\right) \ge c_{ih}, i = 1, 2, ..., n; h, k = 1, 2, ..., m \tag{2}$$

$$c_{jk} - c_{ik} + M\left(1 - x_{ijk}\right) \ge p_{ik}, i, j = 1, 2, ..., n; k = 1, 2, ..., m \tag{3}$$

$$c_{ik} \geq 0, i = 1, 2, \ldots, n; k = 1, 2, \ldots, m \tag{4}$$

Eq.1 represents the objective function, where each inequality constraint represents the priority order of the job sequence for each workpiece, which determines the process constraints and machine constraints, and the priority order of each machine. Under inequality constraint, C_{ik} and P_{ik} are divided by the completion time and the processing time at machine k for part i. M is a sufficient integer, a_{ihk} and x_{ijk} are the indicator coefficients and the indicator variables, and their meaning is as follows.

a_{ihk} 1 When the machine h first machining the part i rather than the machine k,

0 When not described above.

x_{ijk} 1 When the part i is preceded by the part j, the machining at the machine k,

0 When not described above.

3 The Job-shop scheduling program implementation based on genetic algorithm

In this section, it is described a genetic algorithm steps for solving the job-shop scheduling problem, and is proposed a new solution method to obtain the optimal solution using a program based on it.

3.1 Genetic algorithm steps

1) Encoding

Based on the Job-Shop problem characteristics, we choose a coding scheme based on the working sequence (machining sequence, machining time on each machine). This way, schedule encoding is a working sequence, and each gene represents a working sequence. We use the following method for working sequence encoding. The detailed encoding process is described as the following example: For a scheduling problem consisting of three workpieces and three machines, a chromosome is generated based on {2, 3, 1, 3, 1, 2, 3, 2, 1}.

At this time, O_{ij} is the jth working order for workpiece i, there is a rule according to process constraints, and there must be a sequence operation table corresponding to chromosome based on $[O_{21}, O_{31}, O_{11}, O_{32}, O_{12}, O_{22}, O_{33}, O_{23}, O_{13}]$. We present a schedule by conveniently inverting by the coding rule at decoding.

2) Fitness function

The fitness function is as follows.

$$f = \left(\left(MS_{\max} - MS_i \right) \times R_a + R_b \right)^2 / \sum_{p=1}^{n} \left(\left(MS_{\max} - MS_i \right) \times R_a + R_b \right)^2 \tag{5}$$

Where $MS_{\max}$ is the maximum objective function value of individuals in the current population, MS_i is the objective function of individual i, n is the population number in the population, and R_a, R_b are real numbers.

3) Genetic operation

We mainly study with crossover and mutation that affect the algorithm performance. In genetic algorithms, the choice of crossover probability (P_c) and mutation probability (P_m) has a crucial impact on algorithm performance, performance and algorithm convergence. In a general genetic algorithm, when the individual fitness value approaches the maximum fitness value, the crossover probability and mutation probability become smaller, and when the largest fitness value is equal, the crossover probability and mutation probability become zero. This kind of control method for populations is relatively suitable for populations to be located in the late stages of evolution, but is unfavourable in the early stages of evolution. This is because the relatively superior individual in the population of the early stages of evolution is in a state where little change occurs. This increases the feasibility that evolutionary flow is locally best, so we use an improved approach to this problem.

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, f' < f_{avg} \end{cases} \tag{6}$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f - f_{avg})}{f_{\max} - f_{avg}}, & f \geq f_{avg} \\ P_{m1}, f < f_{avg} \end{cases} \tag{7}$$

According to this improved method, when the algorithm is implemented, the crossover probability and mutation probability of the fittest individual can improve P_{c2} and P_{m2}, respectively, and obtain the best population.

Cross ever

Choosing two parent chromosomes Pl and P2 and assuming that the crossover probability satisfies the crossover condition, it can perform crossover operations. The crossover operation uses single-point crossover. Therefore, when set cross point arbitrarily in sequence of individuals and run the cross, the substructure of the two individuals forward or back of this point is compatible and generate two new individuals.

Mutation

The mutation operation randomly exchanges the positions of two different genes in the chromosome. In mutation operations, SWAP mutation operators can increase population diversity during genetic evolution, but due to the nature of Job-Shop problem itself, there may exist situations in which the workpieces of different gene loci are equal (representing different processes of the same workpieces) in the chromosome sequence. Therefore, modifications must be performed when performing the substitution mutation. The detailed method allows judgment of the two-locus gene to be interchanged. If the workpiece number of the two-locus gene is the same, the two-locus gene (random) is selected again. Then, substitution is performed again until the gene of the workpiece number is found.

3.2 Optimal solution using job-shop scheduling program

The main function of the scheduling program is to generate one feasible and optimization working sequence according to the manufacturing requirements based on the waiting part information and the resource state information inside the cell.

3.2.1 Job-shop scheduling program flow

A job-shop scheduling program based on genetic algorithms was developed in Visual Studio.NET 2015 environment. The scheduling program flow chart is shown in Figure 1.

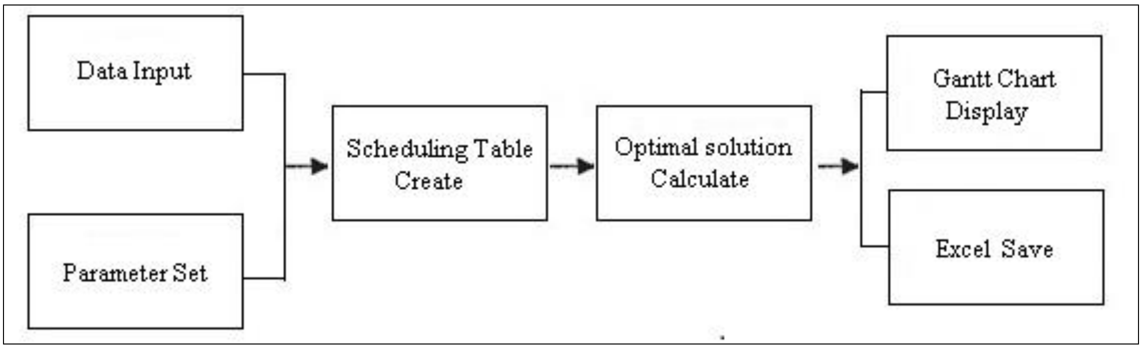


Fig1: The scheduling program flow chart

1) Data Input and Parameter Set

In the data input module inputs the basic data needed for scheduling such as workpiece, machine number, process sequence, and machining time. We schedule with five parts, three machines and four processes. In the basic parameter setting, input the parameters needed for the GA implementation, including population number, evolutionary algebra, and mutation probability. The types of genetic operators used in program are as follows (Figure 2).

Selection Type

Tournament, Roulette Wheel, Uniform, Rank

Crossover Type

Single point, two - point, Uniform, Ordered

Mutation Type

Exchange values, change value, Slip Block, Replacement

With these types of genetic operators, it can find an optimal solution for the scheduling problem (Figure 2). The property of the next generation generated by crossover is determined by mutation. Thus, the property of the child individuals depends on the mutation method, so the scheduling solution in the genetic algorithm can vary depending on the selection of the mutation type. Therefore, we consider an optimal solution according to the choice of mutation type.

Fig 2: Types of genetic operators

2) Create Table, Calculate Optimal Solution, and Display

The scheduling generation module generates an initial schedule based on the input data and processing time. In the optimal solution calculation module, each fitness value is calculated by a genetic algorithm operation according to the selection of the mutation type with an initial scheduling table. Then, by comparing the obtained target values, it finds the optimal values and hence obtains the workpiece scheduling sequence. During the optimal solution process, the makespan and machine idle time are displayed as Gantt diagrams, and the scheduling information generated in the scheduler is stored in Excel to connect with the Flexsim program.

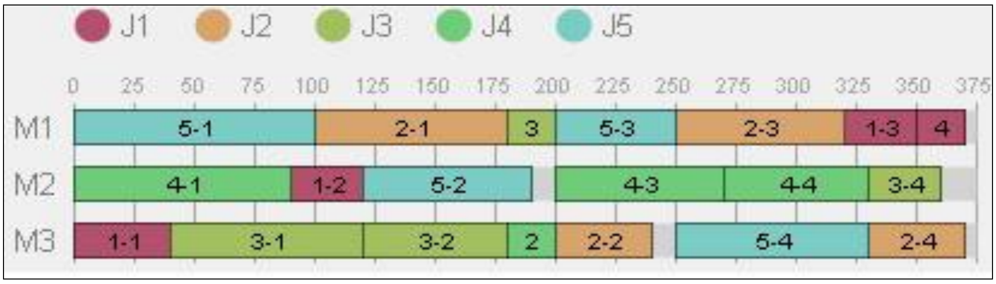
3.2.2 Program execution

The operating parameters of the genetic algorithm are as follows. The population size is 40, the crossover probability is 0.8, the mutation probability is 0.14, the evolution number is 200, R_a is 0.6, and R_b is 100. In genetic operation the selection is Roulette wheel, the crossover uses single-point. First, the program generates initial schedule data based on input data and genetic parameter settings (Figure 3).

	J1			J2			J3			J4			J5		
	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
P1	70	X	40	80	120	X	100	150	80	X	90	50	100	X	150
P2	X	30	X	X	X	40	X	20	60	60	X	20	X	70	140
P3	30	X	60	70	140	X	20	X	40	X	70	120	50	80	X
P4	20	40	X	80	X	40	60	30	X	90	60	30	40	60	80

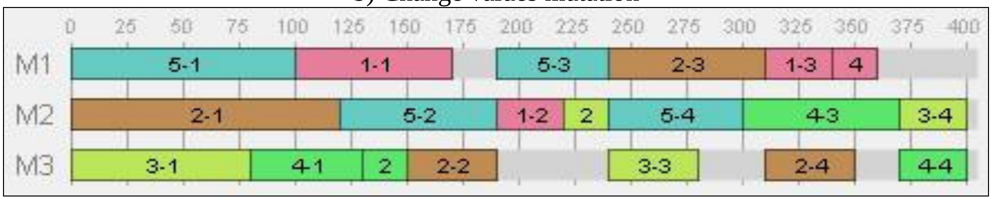
Fig 3: Initial scheduling data

For example, as seen in Figure 3, workpiece 1 is first machined in machine 1 for 70 minutes, then it is machined in machine 3 for 40 minutes, and in the second process it is only machined in machine 2. In the process sequence, in the fourth process, machine 1 is machined for 20 minutes and then in machine 2 is machined for 40 minutes. Then, the program computes the optimal schedule solution according to the mutation type with the initial schedule data and displays the performance of the optimal solution, including makespan and machine idle time in Gantt diagram (Figure 4).

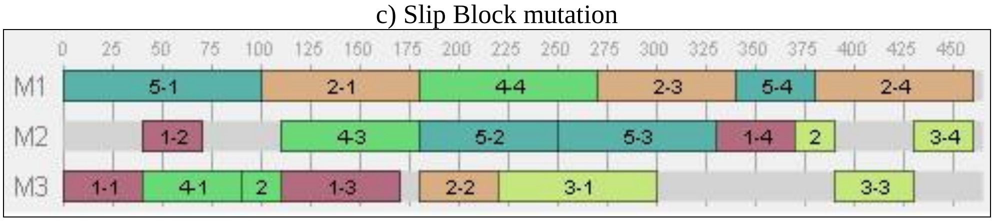


a) Exchange values mutation

b) Change values mutation



c) Slip Block mutation



d) Replacement mutation

Fig 4: Gantt diagram generated by the mutation type As seen in Figure 4, it can know that the makespan and total machine idle time are different according to the mutation type.

4 Simulation of JSSP based on Flexsim

4.1 Flexible Manufacturing System Simulation Model

The discrete event simulation software FlexSim creates and simulates 3D models using a given library and tool, and displays and analyzes the resulting state. We generated the same simulation model as the FMS using FlexSim2022.2.1 (Figure 5).

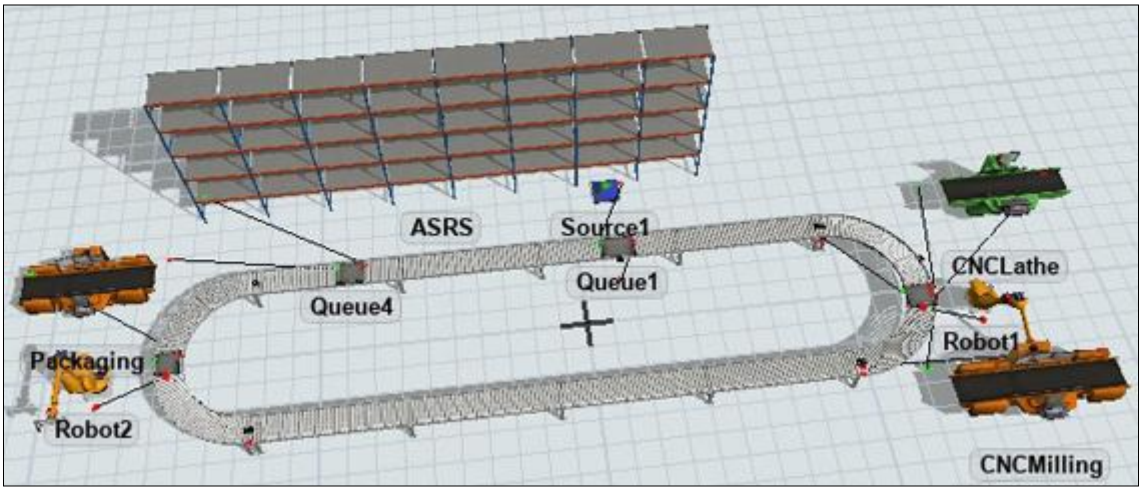


Fig 5: Model layout of the FMS in Flexsim

The simulation model consisted of a ring belt conveyor, a solid storage ASRS for storing and retrieving the workpiece, a loading point and a machining position point for entering the workpiece into the system, an ER9 robot supplying the material to the machine, a CNC lathe and CNC milling machine, a workpiece receiving location and inspection area, and a unloading area. It also includes the ER5 robot responsible for supplying and unloading workpieces to the assembly machine, the packing area and the automatic guided vehicle (AGV). The material flow in the FMS simulation model is as follows:
Five parts are provided as workpieces and each workpiece is machined in CNC lathe and CNC milling machine according to the optimal schedule. After passing through the inspection area, it is assembled in the assembly area, and then stored in the warehouse by AGV at the unloading position.

4.2 Simulation of job-shop scheduling problem

The main objective of the simulation is to verify that the makespan is minimized by analyzing the queue time of the workpieces, idle times of machines, utilization rate of machines, etc. through simulation based on the optimal scheduling information.

4.2.1 Simulation parameters and elements

The entity elements used in the simulation model are seen in Table 1.

Table1: Entity Elements of Simulation Model

Name of elements	Type	Remark
Item1	Item	
Source1	Producer	Produce Item1
Processor1	Processor	CNCLathe
Processor2	Processor	CNCMilling
Processor3	Processor	Packaging
Conveyor1	Conveyor	Material Flow Conveyor
Robot1	Robot	Part Supply Robot
Robot2	Robot	Load/unload Robot
Rack1	Rack	ASRS
Queue1	Queue	place of part receipt
Queue2	Queue	place of part input
Queue3	Queue	place of package input
Queue4	Queue	place of Part unload

4.2.2 Run of simulation

The simulation flow chart proposed in this paper is seen in Figure 6.

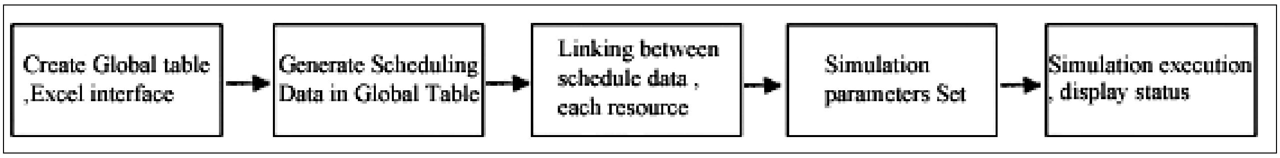


Fig 6: Simulation flow chart

The main idea of the simulation is to generate the schedule table <Global Table> as a combination between the optimal schedule results stored in Excel, and the Flesim. It is also a combination of schedule data and each resource in the model. Then, it simulates the schedule according to the mutation type, with a feasible operating parameter. The simulation method based on job-shop scheduling data in FlexSim is as follows.

Connections between <Global Table > and Excel software

First, add a new Global Table and Excel interface (Excel Import/Export) from the ToolBox's menu bar. Then, it after setting the Excel filename and the <Global Table> table name to connect in the Excel interface window, perform the necessary operations. The schedule information is connected to <Global Table>. Figure7 is <Global Table> that displays the scheduling data generated by the Exchange vales mutation.

model | table1 | GlobalTable1 | GlobalTable2 | GlobalTable3 | GlobalTable4

Part	Machine	Time	Machine	Time	Machine	Time	Machine	Time
1	3	40	2	30	1	30	1	20
2	1	80	3	40	1	70	3	40
3	3	80	3	60	1	20	2	30
4	2	90	3	20	2	70	2	60
5	1	100	2	70	1	50	3	80

Fig 7: Scheduling data based on Exchange values mutation

Connections between <Global Table> and resources in the model

In the property window of each resource within the model, select Flow->Put strategy->By Global Table Lookup and then assign the table name and the schedule data for each resource (processing sequence, processing start time, completion time).

Run of simulation

In system simulation time, the unit is equal to the actual one second. In the simulation, a set of components for running the machines and supplying the workpiece is represented in the system resource (Source). For example, to represent five parts, we have to obtain different color components in <Triggers> on creation> set label and colour>. To obtain five color objects from the source, we use Duniform (1, 5). During the simulation, it visually displays the operating conditions such as each machine utilization rate, idle time, waiting time of workpiece, start and finish time (Figure8).

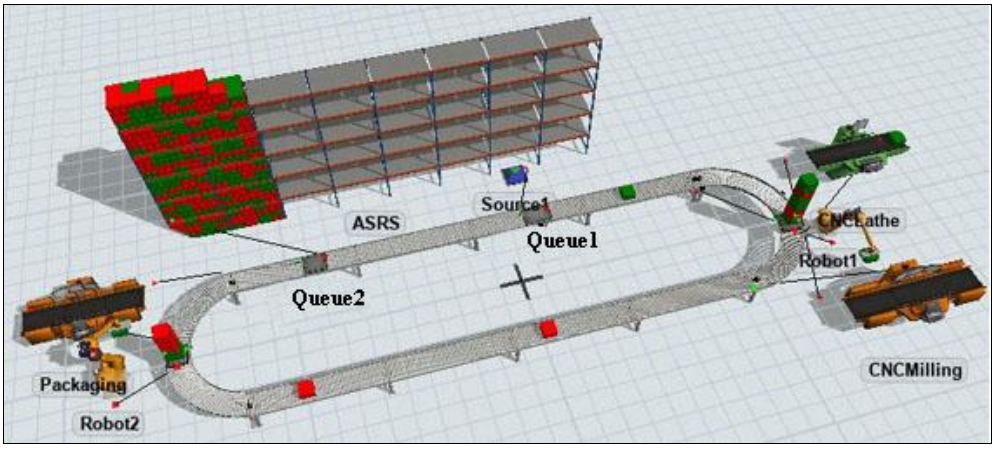


Fig 8: Display of simulation

Also, after the simulation run, the results of each schedule solved by the genetic algorithm are analyzed automatically.

5. Results and Discussion

We computed the optimal solution of the job-shop scheduling problem using a GA-based scheduler. The scheduling program allowed to be obtained the best solution with a makespan of 370 minutes and the machine idle time of 60 minutes, when the Exchange values mutation method was chosen (Table 2). Through computational results, the new solution method with the selection of mutation type seems to be an efficient method for solving job-shop scheduling problem with minimizing the makespan (Figure 9).

Table 2: Solution results

Mutation type	Makespan(min)	Idle time(min)
Exchange values	370	60
Slip Block	400	160
Replacement	460	250
Change value	490	360

Also, we compared and analyzed the results by performing simulations separately with the scheduling data solved according to the mutation type. From the results, the machine utilization rate seems to be relatively high when simulated with the optimal schedule obtained by the Exchange values mutation method (Figure10). As seen in Figure 10, the utilization rates of the lathe and milling machine are different. This is because the machining time required for the two machine tools is different. Besides, bottlenecks occur at the workpiece feed position. This is because two machine tools share one robot, so the robot cannot meet the transportation requirements in time. To reduce the bottleneck, the workpiece that requires only packaging among the workpiece can be increased accordingly. Simulation using FlexSim allowed us to verify our proposed optimal solution method and to evaluate the performance of the system by knowing the process states, such as machine utilization rate, average waiting time of jobs, and average idle time for each machine.

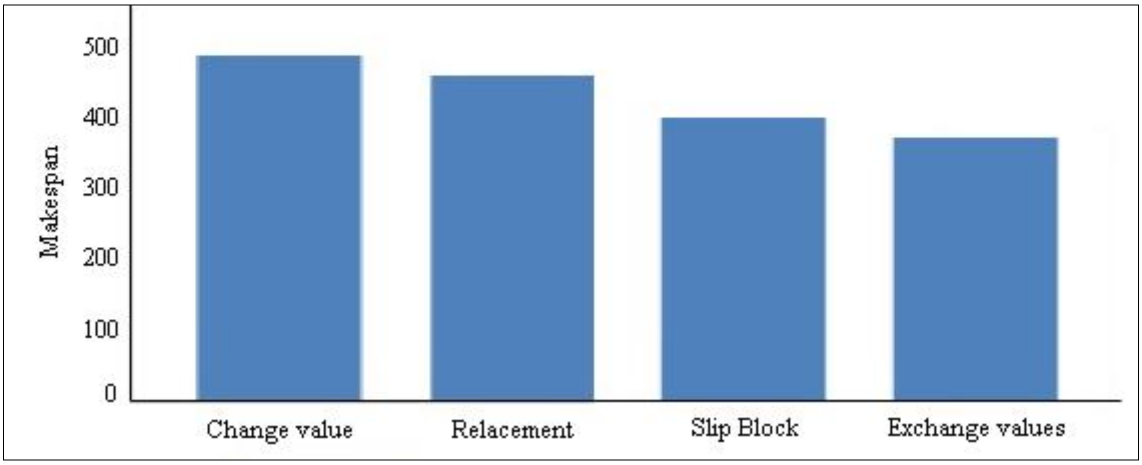
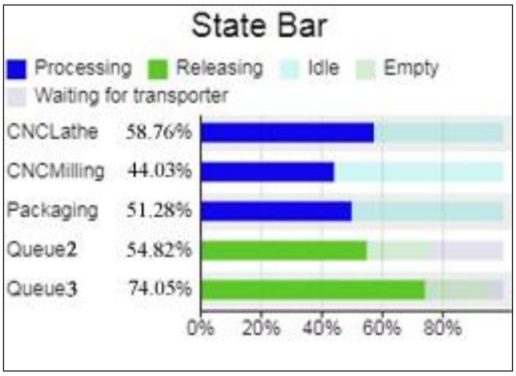
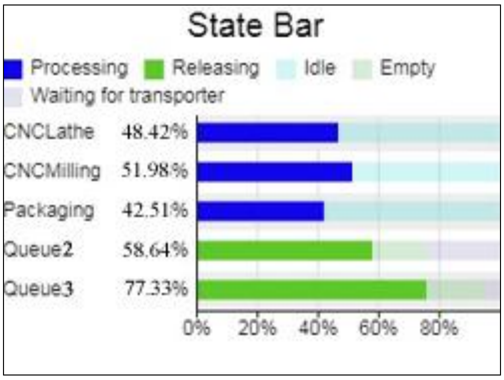


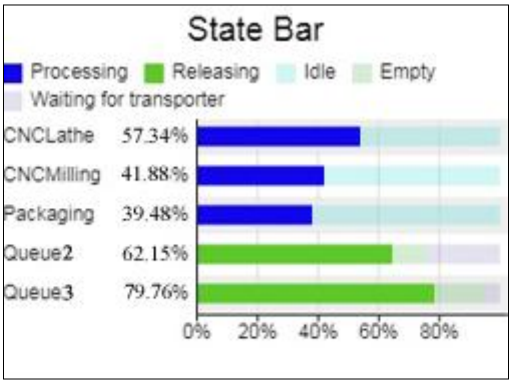
Fig 9: Comparison of makespans



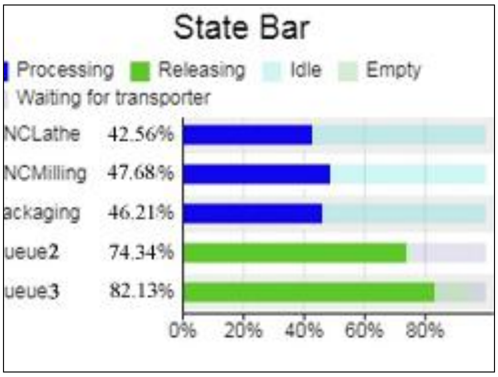
a) Exchange value mutation



b) Slip Block mutation



c) Replacement mutation



d) Change value mutation

Fig 10: Bar chart representing the utilization rate of machines

We are aware that our research may have some limitations. In genetic algorithm, the genetic operators including selection of crossover method may affect the optimal solution according to any object. Therefore, when finding the optimal solution according to any object in a scheduling program, we can use selection of crossover type including mutation type.

6. Conclusion

This paper has constructed a mathematical model with makespan minimization as an objective function in a job-shop scheduling problem and has applied a genetic algorithm-based job-shop scheduling program. We have proposed a new optimal solution method according to the selection of mutation type in job- shop scheduling program. The solution result was optimized the total processing time by 160 min of makespan determined by the Exchange values mutation technique. From the results, it demonstrated that the new solution method according to the selection of mutation type has a good advantage in solving job-shop scheduling problem. Also, the job-shop results obtained by selection of mutation type have simulated using Flexsim software. The simulation results showed that the proposed method can reduce the makespan, increase the machine utilization rate and realize the process optimization. The proposed method can be applied to the optimal scheduling solution to minimize the makespan in manufacturing systems, including flexible manufacturing systems. In future work, we will add a combinatorial selection algorithm of crossover type and mutation type to improve the obtained optimal scheduling solution. This technique can be embedded in a genetic algorithm to obtain a better optimal solution according to any object.

Acknowledgment
This work was supported by Kim Chaek University of Technology, Democratic People's Republic of Korea. The supports are gratefully acknowledged. The authors express their gratitude to the editors and the reviewers for their helpful suggestions for improvement of this paper.

References
1. Anshulika. Application of genetic algorithm on job shop scheduling problem to minimise makespan. International Journal of Science and Research. 2016;5(6).
2. Zacharia PT. AGV routing and motion planning in a flexible manufacturing system using a fuzzy-based genetic algorithm. The International Journal of Advanced Manufacturing Technology. 2020;—.
3. Salido MA. A genetic algorithm for energy-efficiency in job-shop scheduling. Int J Adv Manuf Technol. 2016;85:1303-1314.
4. Bao Bizhen. A new algorithm of the scheduling of an FMS based on genetic algorithm. Manufacturing Rev. 2023;10:11.
5. Wu Rui. Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. Advances in Mechanical Engineering. 2018;10(10):1-14.
6. Rohaninejad M. A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem. Int J Adv Manuf Technol. 2015;77:51-66.
7. Zhou Erming. Flexible job-shop scheduling based on genetic algorithm and simulation validation. MATEC Web of Conferences. 2017;100(GCMM2016):—.
8. Habbadi S. Optimization of job shop scheduling problem by genetic algorithms. Management and Production Engineering Review. 2023;14(3):44-56.
9. Habbadi S. Job shop scheduling problem using genetic algorithms. Proceedings of the European IEOM Rome Conference, University Sidi Mohammed Ben Abdellah. 2022;—.
10. Prathipatil N. Flexible job-shop scheduling using software. International Journal of Engineering and Advanced Technology. 2020;9(3):—.
11. Yadav A. Makespan minimization in flexible manufacturing system using genetic algorithm. International Journal of Applied Engineering Research. 2018;13(11):9328-9334.
12. Dagroum AAO. Genetic algorithm for minimizing the makespan in job shop scheduling with machine setup. International Research Journal of Engineering and Technology. 2021;8(2):—.
13. Meng L. A novel hybrid algorithm of genetic algorithm, variable neighborhood search and constraint programming for distributed flexible job shop scheduling problem. International Journal of Industrial Engineering Computations. 2024;15:813-832.
14. Raju D. Optimization of JSS problem using FlexSim simulation software. International Journal of Mechanical Engineering and Technology. 2018;9(5):299-305.
15. Krynke M. Personnel management on the production line using the FlexSim simulation environment. Manufacturing Technology. 2021;21(5):—.
16. Pontes R. Machining process optimization using simulation in FlexSim. Proceedings of the North American International Conference on Industrial Engineering and Operations Management, Orlando, Florida, USA. 2022;—.
17. Gopinath G. Optimization of routing and scheduling layout by using FlexSim. IJARIIE. 2023;9(1):—.
18. Kumar BS. Modeling and analysis of flexible manufacturing system with FlexSim. International Journal of Computational Engineering Research. 2015;5(10):—.
19. Kliment M. Creation of simulation models using the FlexSim software module. Advanced Logistic Systems - Theory and Practice. 2022;16(1):41-50.