International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 9, pp. 3549-3565
doi: 10.25147/ijcsr.2017.001.1.231
https://stepacademic.net

**Short Paper**

# An Improved Genetic Algorithm for Solving the Multi-Objective Flexible Job Shop Scheduling Problem

Lei Chen
Graduate School, University of the East Manila, Philippines
554535432@qq.com
(corresponding author)

Joan P. Lazaro
Graduate School, University of the East Manila, Philippines
joan.lazaro@ue.edu.ph

*Date received*: September 11, 2024
*Date received in revised form*: January 13, 2025
*Date accepted*: January 26, 2025

Recommended citation:

Chen, L., & Lazaro, J. P. (2025). An improved genetic algorithm for solving multi-objective flexible job shop scheduling problem. *International Journal of Computing Sciences Research*, 9, 3549-3565. https://doi.org/10.25147/ijcsr.2017.001.1.231

**Abstract**

*Purpose* – Aiming at the flexible job shop scheduling problem (FJSP), a multi-objective scheduling model with the maximum completion time and the minimum total processing energy consumption was constructed as the optimization objectives.

*Method* – To optimize the performance of the genetic algorithm, a diversified population strategy was adopted, combining elite retention and a random roulette wheel selection mechanism to screen and retain offspring with excellent performance carefully At the operational level of the algorithm, an efficient single-point crossover strategy was adopted for the process coding part to promote the combination and transmission of excellent characteristics. For the machine allocation part, a flexible random crossover method was introduced to increase the diversity of solutions. To explore new solution spaces and avoid premature convergence, a single-point mutation mutation strategy based on minimum machine energy consumption was designed.

*Results* – Simulated the workshop manufacturing of six machines and six job positions in a company. Compared to the original scheme, the optimized one reduced the production interval by 19.67%, proving its effectiveness.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

*Conclusion* – There are other important objectives in actual production, and dual objective optimization may not fully reflect the complexity and diversity of the production system. In addition to dual objective optimization, considering multi-objective optimization methods may better capture the authenticity of the production system comprehensively.
*Recommendations* – By optimizing production scheduling, overall production efficiency can be improved and profit margins can be increased.

*Practical Implications* – This study provides a solution to improve production efficiency and reduce energy consumption in manufacturing environments. The proposed method has been validated through enterprise cases, demonstrating its effectiveness in optimizing maximum completion time and total energy consumption. Implementing this algorithm can improve profit margins and promote more sustainable production practices, aligning with the goals of modern manufacturing.

*Keywords* – FJSP, genetic algorithm, multi-objective, multiple populations

## INTRODUCTION

With the continuous expansion of production scale, effectively arranging various resources has become the main task of digitization in the current manufacturing industry. Production scheduling is the core of manufacturing planning, and in-depth research on it is beneficial for energy conservation, emission reduction, and efficiency improvement, thereby accelerating the transformation and upgrading of the manufacturing industry. The flexible job shop scheduling problem (FJSP) is a further extension of the traditional job shop scheduling problem (JSP), which has been proven to be an NP-hard problem. This problem allows each process to be processed on any available machine. In addition to determining the sequence of process processing, it is also necessary to determine the corresponding relationship between specific processes and machines, making it more complex compared to JSP. However, in actual production processes, there is a demand for multiple tasks to be carried out simultaneously and to be arranged in a reasonable order to improve production efficiency and resource utilization. Due to the interdependence between different tasks and conflicts between multiple optimization objectives, such as minimizing production time, maximizing profits, minimizing equipment idle, etc., solving the multi-objective job shop scheduling optimization problem becomes very complex.

## LITERATURE REVIEW

### *Multi-objective Flexible Job-shop Scheduling*

In recent years, with the introduction of various algorithms and optimization technologies, the field of multi-objective FJSP has made significant progress. Xia and Wu (2005) proposed a hybrid optimization method using simulated annealing (SA) as a local search algorithm and demonstrated its effectiveness for large-scale FJSP. Moslehi and Mahnam (2011) developed a Paretobased method that combines particle swarm optimization and local search, demonstrating competitiveness in solving multi-objective FJSP problems. Li et al. (2011) proposed a hybrid Paretobased discrete artificial bee colony algorithm for FJSP and added a fast Pareto set update function. Piroozfard et al. (2018) introduced an improved multi-objective genetic algorithm for minimizing the total carbon footprint and total delay working standard in FJSP. Wang (2020) proposed a hybrid multi-objective evolutionary algorithm based on decomposition to solve the FJSP problem under the time-sharing price, aiming at minimizing the completion time and the total power cost simultaneously. Liang et al. (2021) proposed an improved adaptive non-dominated sorting genetic algorithm, which has an elite strategy and is used to solve the multi-objective FJSP problem, and its effectiveness was proved by simulation results. Luo et al. (2021) explored the dynamic multi-objective scheduling of flexible job shops through deep reinforcement learning, emphasizing the application of advanced learning techniques in scheduling optimization. Wei et al. (2023) proposed a multi-objective migratory bird optimization algorithm based on game theory for dynamic FJSP problems, demonstrating the integration of natural heuristic algorithms and strategic decision-making concepts. Du et al. (2023) proposed a knowledge-based reinforcement learning and distribution estimation algorithm for FJSP, which combines deep Q networks with domain knowledge to effectively improve scheduling solutions. Shi and Xiong (2024) proposed a multi-objective job shop scheduling problem considering total delay (MOJSSP/O) and developed an enhanced non-dominated sorting genetic algorithm II (ENSGA-II) to solve the problem.

### *Genetic Algorithm*

Genetic algorithms have been widely used in various fields for optimization and problem-solving purposes. Chapman et al. (2021) stated that the genetic algorithm is a search and optimization technique based on the theory of natural selection, commonly applied to various complex problems. Mantawy et al. (1999) presented a new algorithm that integrates genetic algorithms, tabu search, and simulated annealing to solve the unit commitment problem. The core of this algorithm is genetic algorithms, with tabu search used for population generation and simulated annealing employed to accelerate convergence. Slowik and Kwasnicka (2020) presented genetic algorithms as one of the evolutionary algorithms that can be implemented in any programming language. Yin et al. (2024) proposed an adaptive genetic algorithm that considers the overall evolutionary state of the population. By dynamically adjusting the crossover and mutation probabilities, the algorithm effectively improves the global search ability and convergence efficiency of the genetic algorithm.

## *Genetic Algorithm for Job Shop Scheduling*

Gu et al. (2010) proposed a competitive coevolutionary quantum genetic algorithm for stochastic job shop scheduling problems, aiming to minimize the expected completion time. This algorithm utilizes multiple swarm methods and quantum theory concepts to improve diversity and convergence speed. Xing et al. (2011) also proposed a multi-population interactive collaborative evolutionary algorithm for flexible job shop scheduling problems, which combines ant colony optimization and genetic algorithm to achieve independent evolution of the population. Fakhrzad et al. (2013) proposed a multi-objective job shop job scheduling method using a hybrid genetic algorithm. This method combines advantage relationship and weighted aggregation fitness calculation, introducing fitness-based advantage relationship and weighted aggregation in genetic algorithm and local search, respectively. This method effectively improves scheduling efficiency and has become an important progress in this field. Jiang and Le (2014) focused on the multi-objective flexible job shop scheduling problem considering energy consumption and developed an improved non-dominated sorting genetic algorithm. Tan et al. (2015) used a multi-objective evolutionary algorithm for job shop scheduling, specifically, an improved micro genetic algorithm based on the Pareto optimality principle was adopted. X. Zhang et al. (2020) proposed a hierarchical multi-strategy genetic algorithm based on non-dominated sorting for optimizing energy efficiency in integrated process planning and scheduling. Xie et al. (2023) proposed the Hybrid Genetic Taboo Search Algorithm (HGTSA), which effectively combines global capabilities (GA) and local capabilities (TS). To evaluate HGTSA, it was compared with four state-of-the-art algorithms. The experimental results indicate that it outperforms these comparisons in terms of solution quality and computational efficiency. Reijnen et al. (2023) proposed an automated deep reinforcement learning (DRL) method specifically aimed at online control of multiple objectives. When applying the genetic algorithm (GA) to the FJSP for testing, the results indicated that DEMOCA's performance was comparable to that of the grid search, while it significantly reduced the required training cost. Momenikorbekandi and Abbod (2023) devised a novel hybrid Parthenogenetic Algorithm (NMHPGA) to optimize the flexible single- and multi-machine shop furnace process. The comparative results demonstrate that NMHPGA achieves a superior objective function value at a faster rate.

## *Energy consumption for job shop Scheduling*

Tang and Dai (2015) proposed an energy-saving method to minimize energy consumption in the extended job shop scheduling problem, with a focus on modifying the scheduling of machines working at different speeds in the job shop. Lei et al. (2018)

developed a two-phase meta-heuristic for multiobjective flexible job shop scheduling with a total energy consumption threshold. Yüksel et al. (2020) solved an energy-saving dual objective permutation flow shop scheduling problem to minimize total delays and energy consumption, emphasizing the necessity of sustainable production practices. Zuo et al. (2023) studied an assembly mixed flow shop scheduling problem with energy consumption and proposed an artificial bee colony optimization algorithm based on population diversity. Yu et al. (2024) proposed a co-evolutionary algorithm based on deep Q-learning networks to solve the NP-hard problem of minimizing total energy consumption (TEC) and makespan. An efficient heuristic method can reduce TEC.

According to the current research status of FJSP by scholars both at home and abroad, several issues have been identified: (1) The optimization algorithm used has average optimization ability, mainly due to the algorithm's long response time, failure to consider the initial population screening situation, easy entry into local optimal, and the inability to effectively dynamically adjust key parameters, resulting in solution efficiency and quality that cannot meet production requirements; (2) The optimization objectives are often too narrow, frequently focusing only on minimizing the maximum completion time, without adequately reflecting the actual production conditions of the job shop. Therefore, this article establishes a mathematical model aimed at minimizing both maximum completion time and energy consumption and proposes an improved algorithm.

## METHODOLOGY

### *Problem Description and Mathematical Modeling*

FJSP can be described as a machining system with M machines that require processing N jobs, where the number of processes included in job k is not fixed. For the convenience of expressing FJSP using mathematical models, Table 1 provides definitions for the parameters.

Assuming that the job is not allowed to be interrupted during the processing, the machine remains in a standby state at time zero. It is stipulated that the processing process needs to meet the following constraint relationships:

1) There is a sequential constraint relationship between the processes of the same job.

2) Each machine can process any process, but each machine can only process one process at a time.

3) The job can only be processed on one machine at a time, and the processing status cannot be interrupted.

4) After the current process is completed, the transportation time consumed during the process is not calculated.

Table 1. Definition of relevant parameters

| Parameter | Definition |
|---|---|
| M | Number of machines |
| N | Quantity of jobs |
| i | Job Number |
| j | Operation sequence |
| k | Machine No |
| $O_{ij}$ | The j-th process of job i |
| $t_{ijk}$ | The machining time of the j-th process of job i on machine k |
| $S_{ijk}$ | The j-th process of job i starts at the time of machine k |
| $F_{ijk}$ | The completion time of the j-th process of job i on machine k |
| $C_i$ | Completion time of job i |
| $C_{max}$ | Makespan |
| E | Total energy consumption of machine processing |
| $e_{ijk}$ | The machining energy consumption of the j-th process of job i on machine k |
| $y_{ijk}$ | Can machine k process the jth step of the job;i$y_{ijk}$=1, Yes; $y_{ijk}$=0, No |
| α | The weight of the maximum completion time in the scheduling plan |
| β | The weight of machine processing energy consumption in the scheduling plan |

This paper aims to establish a multi-objective FJSP model. The objective function is to minimize the maximum completion time and energy consumption. The scheduling model is as follows:

$$\mathrm{C}_{\max} = \min(\max(C_i)) \qquad (1)$$

$$E = \min(\sum_{i=1}^{N}\sum_{j=1}^{N_i}\sum_{k=1}^{M} t_{ijk} y_{ijk} e_{ijk}) \qquad (2)$$

$$S_{ijk} - S_{i(j-1)k'} \geq t_{ijk} \qquad (3)$$

$$S_{ijk} + t_{ijk} \leq F_{ijk} \qquad (4)$$

$$t_{ijk} > 0 \qquad (5)$$

$$S_{ijk} \geq 0 \qquad (6)$$

The processing of the objective function, weighted:

$$f(x) = \min(\alpha \times C_{\max} + \beta \times E) \quad (7)$$

Equation (1) represents minimizing the maximum completion time; Equation (2) represents minimizing processing energy consumption; Equation (3) represents the sequential constraint relationship of the same job process; The completion time of any process must meet the constraint of equation (4); Equation (5) represents that the processing time of any process must be greater than zero; Equation (6) indicates that all jobs need to start processing at time 0; Equation (7) is the objective function.

## *Design of Improved Genetic Algorithm*

In the research on the FJSP problem, the traditional genetic algorithm has a slow convergence speed and is prone to premature convergence. This article focuses on the scheduling problem of flexible job shops and proposes improvements to the genetic algorithm. The improved algorithm flowchart is shown in Figure 1, and the specific steps are as follows:

Step 1: Set relevant parameters and randomly generate multiple initial populations.
Step 2: Encoding and Decoding.
Step 3: Adopt elite retention and random roulette strategies to select outstanding individuals.
Step 4: Calculate the individual fitness values for the population.
Step 5: Perform single-point crossover on the selected parent process section and random crossover on the machine section.
Step 6: Apply single point mutation with minimum machine energy consumption to the selected parent machine portion.
Step 7: Determine whether the termination condition has been met. If the maximum number of iterations or the stopping condition has been reached, terminate the iteration process. Otherwise, return to step four.
Step 8: Output the results.

### *1. Initialization*

In the framework of evolutionary algorithms, the importance of population initialization as the starting step is self-evident, as it is directly related to the efficiency and effectiveness of genetic algorithms in solving complex problems. The complexity of FJSP is not only reflected in the need to allocate the most suitable machine for each task but also in the reasonable arrangement of task sequences, namely process sorting. Currently, most studies tend to adopt a random initialization strategy to construct the initial population. While this approach is simple and feasible, it often leads to uneven quality of the initial solution set and uneven workload distribution among machines. This will cause the algorithm may need to explore better solutions by increasing the number of iterations or expanding the population size, which significantly increases the time cost of the entire optimization process. According to Kacem et al. (2002), this can lead to a substantial increase in the overall time cost of the optimization process.

In multi-population algorithms, three initial populations are randomly generated first, with the size of each population defined independently. The three populations undergo independent evolutionary operations. Due to the randomness of genetic evolution, the three populations may have different results throughout the evolutionary process, but they can provide valuable insights for determining the optimal solution. G. Zhang et al. (2020) discussed the initial population initialization strategy and found that three distinct types of initial populations can effectively optimize the workload allocation of each selected machine in the initial solution, thereby ensuring maximum machine utilization and minimizing the maximum completion time.

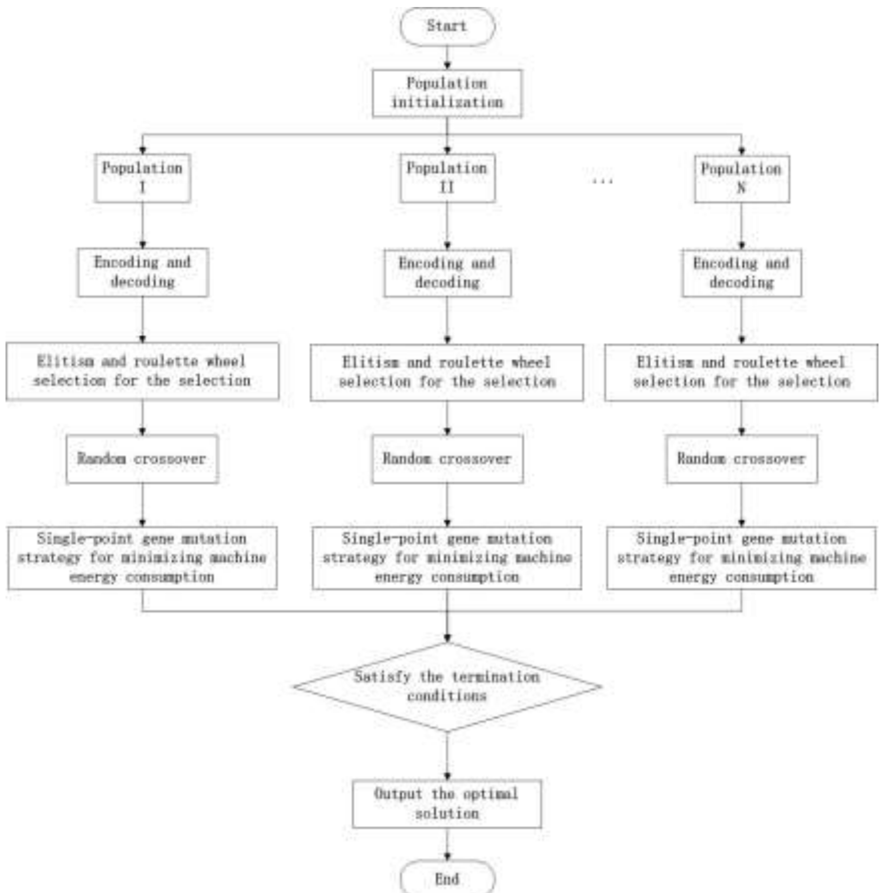


*Figure 1. Flow chart of the improved genetic algorithm.*

## 2. *Encoding and Decoding*

Transforming the solution of the scheduling problem into a chromosome solution capable of genetic operations is called encoding, which is the key to genetic algorithms. This article employs segmented encoding, first to encode the machines to determine the processing machine for each process, and then to encode the processes to determine the order of processing on each machine. Figure 2 shows an example of encoding from Table 2, which represents the solution to the optimization problem. If encoded as 13231 12121. So the processing machines corresponding to the steps O11 → O21 → O12 → O22 → O13 are M1 → M3 → M2 → M2 → M1.

Table 2. Processing Time Table of an Instance of FJSP

| Job | Operations | M1 | M2 | M3 |
|---|---|---|---|---|
| J1 | O11 | 2 | - | 3 |
| | O12 | 3 | 2 | 2 |
| | O13 | - | 2 | 2 |
| J2 | O21 | 3 | 3 | 3 |
| | O22 | 3 | 4 | - |

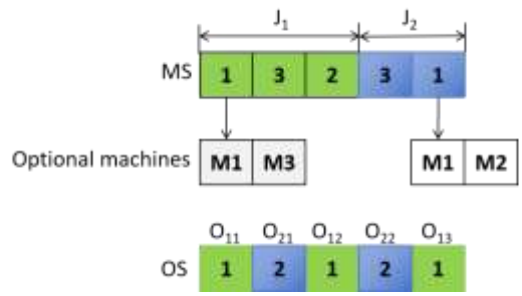


*FIGURE 2. Chromosome coding scheme.*

## *3. Fitness evaluation*

The fitness function is a method for measuring individual quality, screening strategies, and identifying high-quality individuals in a population. It is typically calculated based on the reciprocal of the objective function value. In genetic algorithms, the role of the fitness function is particularly crucial, as it can quantify an individual's adaptability and serve as the basis for selection and evolution. By evaluating fitness, genetic algorithms can optimize the search process, and retain and cultivate outstanding individuals in the population. The fitness function is:

$$fit(f(x)) = \frac{1}{\alpha \times C_{\max} + \beta \times E} \tag{8}$$

In the formula: *fit (f (x))* is the fitness value; *Cmax* is the maximum completion time; *E* is the total load of machine processing; *α. β* is the weight coefficient.

## *4. Selection*

To develop genetic algorithms in better and more advantageous directions in the early stages of iteration, selection operations are used to select outstanding individuals from the population. In our method, the criteria for selecting chromosomes to be included in the mating pool can be chosen from two well-known selection methods in genetic algorithm literature: tournament selection and roulette wheel selection, as shown in Figure 3.

```
function choice_index = choice(fitness, k, pool)
  % fitness: 适应度，根据适应度做锦标赛
  % k: 每次比较的个数
  % pool: 交叉池大小
  n = length(fitness);
  choice_index = zeros(1, pool); % Preallocate the choice_index array

  for i = 1:pool
    random_indices = randperm(n, k);
    f_values = fitness(random_indices);
    [~, min_index] = min(f_values);
    choice_index(i) = random_indices(min_index);
  end
end

function choice_index = choice_lunpandu(fitness, pool)
  % fitness: 适应度
  % pool: 交叉池大小
  n = length(fitness);
  total_fitness = sum(fitness);
  choice_index = zeros(1, pool); % Preallocate the choice_index array
  for i = 1:pool
    % 根据适应度值计算选择概率
    prob = fitness / total_fitness;
    % 使用轮盘赌选择个体
    rand_num = rand;
    cumulative_prob = 0;
    for j = 1:n
      cumulative_prob = cumulative_prob + prob(j);
      if rand_num <= cumulative_prob
        choice_index(i) = j;
        break;
      end
    end
  end
end
```

*Figure 3. Tournament and roulette wheel code.*

## *5. Crossover*

The global search capability of genetic algorithms largely depends on the crossover operator. In traditional genetic algorithms, most crossover operators use two parents to generate offspring, which is a simple method but prone to convergence too quickly. This article focuses on a random selection of chromosome parts, with specific codes shown in Figure 4.

## *6. Mutation*

In the field of traditional genetic algorithms, a common approach is to use a fixed mutation probability for operation. However, this approach may trap the algorithm in local optima, making it difficult to find a better solution. To solve this problem, the machine part can adopt a single-point gene mutation strategy that minimizes machine energy consumption, in order to search for the optimal solution more accurately.

```
function [afinal, bfinal] = cross_MS(A, B)
    job_num = length(A);
    while true
        random_numbers = randperm(job_num, 2);
        if random_numbers(1) ~= random_numbers(2) % 随机选择两个不同的索引
            break;
        end
    end
    rl = min(random_numbers(1), random_numbers(2));
    rr = max(random_numbers(1), random_numbers(2));
    %排除最小值等于1，最大值的最后一个索引
    if rl==1
        rl=rl+1;
    end
    afinal = [B(1:rl-1), A(rl:rr-1), B(rr:end)];
    bfinal = [A(1:rl-1), B(rl:rr-1), A(rr:end)];
end

function [afinal, bfinal] = cross_OS(A, B)
    job_id = unique(A);
    while true
        num_to_extract = randi([1, length(job_id) - 1]);
        if num_to_extract > 0 && num_to_extract < length(job_id)
            break;
        end
    end
    random_index = randi(numel(job_id));
    S1 = job_id(random_index);
    target_indices_A = A == S1;
    elements_to_fill = B(B ~= S1);
    afinal = A;
    afinal(target_indices_A) = S1;
    fill_index = 1;
    for i = 1:numel(afinal)
        if afinal(i) ~= S1
            afinal(i) = elements_to_fill(fill_index);
            fill_index = fill_index + 1;
        end
    end
    target_indices_B = B == S1;
    elements_to_fill = A(A ~= S1);
    bfinal = B;
    bfinal(target_indices_B) = S1;
    fill_index = 1;
    for i = 1:numel(bfinal)
        if bfinal(i) ~= S1
            bfinal(i) = elements_to_fill(fill_index);
            fill_index = fill_index + 1;
        end
    end
end
```

*Figure 4. MS and OS crossover code.*

## SIMULATION RESULTS AND ANALYSIS

To verify the effectiveness of the algorithm designed in this paper, the job shop manufacturing of an enterprise is investigated. The process of processing a product in this enterprise can be simplified into the FJSP problem of six machines processing six jobs. The constraints for processing jobs and machines are shown in Table 3. The first column in the table represents six jobs, the second column represents the process of each job, and columns 3 to 8 represent the machines available for processing, along with their corresponding processing times and energy consumption.

The experimental parameters are set as follows: population size of 500, crossover probability of 0.85, mutation probability of 0.06, and maximum iteration count of 300. The test results under different weight combinations are shown in Table 4.

Table 3. Data of FJSP

| Job | Operations | Processing time/Unit time processing energy consumption | | | | | |
|---|---|---|---|---|---|---|---|
| | | M1 | M2 | M3 | M4 | M5 | M6 |
| J1 | O11 | 10/3 | 15/2.5 | - | 14/2.7 | - | 14/1.5 |
| | O12 | - | 6/3 | 4/2.1 | 16/2.5 | 15/3 | - |
| | O13 | - | 5/2.4 | - | - | 16/2.5 | 8/1.7 |
| | O14 | 12/2 | - | 13/2 | 13/2 | - | - |
| | O15 | 6/3.1 | 6/3.1 | 8/1.5 | 8/2 | 4/2.1 | 9/1.4 |
| | O16 | - | - | 16/1.8 | - | 13/2 | 16/1.6 |
| J2 | O21 | 15/2.5 | - | 6/1 | 16.5/2.2 | 11/2.5 | - |
| | O22 | - | 15/2 | 10/1.5 | 7/2.1 | - | 12/1.5 |
| | O23 | 5/2.3 | - | 16/1.8 | 10/2.3 | 14/2.3 | - |
| J3 | O31 | 14/2.9 | 15/2.7 | 6/1.7 | 5/2.8 | 4/3.1 | - |
| | O32 | - | 5/2 | 6/2 | - | 16/2.2 | - |
| | O33 | 5/2.8 | 8/2.3 | - | 11/2.9 | - | 15/2 |
| | O34 | - | 6/2.5 | 17/2.1 | 14/2.5 | 12/2 | - |
| | O35 | 17/2.5 | - | 14/2.4 | 7/2 | 7/2.8 | 11/2.1 |
| J4 | O41 | 20/3 | - | 19/2 | 13/2.5 | 15/2.3 | - |
| | O42 | - | 10/2.5 | 7/1 | 14/2.8 | 7/3 | 15/2 |
| | O43 | 4/2.8 | 8/2 | - | - | - | 16/2 |
| | O44 | 9/3.2 | - | 6/1.6 | - | 6/2.5 | - |
| | O45 | 16/2.9 | 9/2.3 | 16/2.1 | 13/2 | - | - |
| J5 | O51 | - | 6/2.2 | - | 7/2.8 | 12/2 | 8/0.9 |
| | O52 | 8/3 | - | 12/2.7 | 16/2.7 | - | 6/1 |
| | O53 | 13/2.5 | 12/2.3 | - | - | 16/1.8 | 8/0.7 |
| | O54 | - | 4/2 | 6/2 | 5/3 | 12/1.5 | - |
| | O55 | 13/3.2 | - | - | 8/2.5 | - | 9/1.2 |
| | O56 | 11/2.8 | 3/2.5 | 10/2.5 | 12/2.4 | 16/2 | 5/0.9 |
| J6 | O61 | - | 11/1.2 | - | - | 7/1.9 | 8/0.8 |
| | O62 | - | - | 8/2 | 12/3 | - | 6/1 |
| | O63 | 10/2.4 | 5/1 | - | 13/2.1 | 6/2 | - |
| | O64 | 16/2.2 | - | 8/1.7 | - | - | 12/1.1 |

Table 4. Scheduling scheme results under different weight coefficients

| e.g. | α | β | $C_{max}$ | E |
|---|---|---|---|---|
| 1 | 1 | 0 | 49 | 388.4 |
| 2 | 0.8 | 0.2 | 54 | 378.2 |
| 3 | 0.5 | 0.5 | 61 | 363.5 |
| 4 | 0.3 | 0.7 | 72 | 362.9 |
| 5 | 0 | 1 | 87 | 354.8 |

From Table 2, it can be seen that the maximum completion time and energy consumption values obtained under five different weight schemes differ and are inversely proportional to the corresponding weight values. Specifically, as the weight value increases, the corresponding target value decreases, resulting in a better outcome. Meanwhile, the preferences of decision-makers are crucial, as different preferences can lead to different target values and scheduling plans. Therefore, decision-makers can select appropriate scheduling plans based on the actual production situation. Figure 5 shows the scheduling scheme obtained by the algorithm in this article, with completion time and energy consumption weights of 1 and 0, respectively. Among them, the vertical axis represents the processing machine, and the horizontal axis represents time.

In Figure 5, the original scheduling plan for producing this type of part has a period of 61 minutes, while the optimal scheduling plan, optimized using this algorithm has a period of 49 minutes. Compared to the original scheduling scheme, the optimized scheduling scheme reduced the production interval by 19.67%, thus verifying the effectiveness of the proposed algorithm.

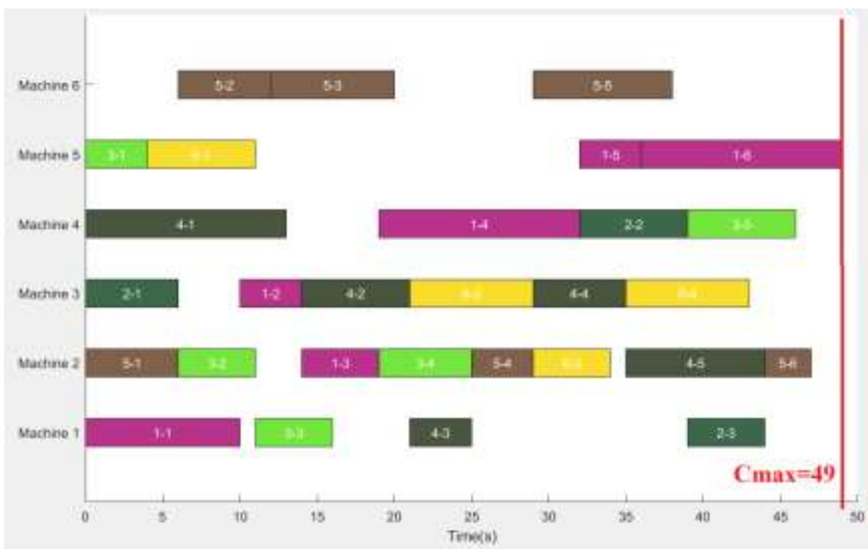

*Figure 5. Gantt chart on the weight of 1,0.*

## CONCLUSIONS AND RECOMMENDATIONS

In this paper, a FJSP problem model is proposed. The objective function is to minimize both the maximum completion time and the total energy consumption. In order to effectively solve this problem, an improved genetic algorithm method is introduced. MATLAB is used to simulate the simplified actual FJSP, and scheduling schemes based on different preferences of decision-makers are presented, along with the corresponding scheduling Gantt chart.

However, this study still has some limitations, as it only considers two objectives: completion time and energy consumption. In actual production, there may be other important objectives, such as equipment utilization, product quality, worker satisfaction, and material flow costs. Single or dual-objective optimization may not fully reflect the complexity and diversity of production systems.

## IMPLICATIONS

In the face of global competition in the integration of the world economy, for the manufacturing industry to stand out in the cruel survival of the fittest, enterprises must accelerate their response speed to external changes, improve product quality and performance, reduce various costs in process links, and provide personalized services according to customer needs. Among these key points, improving the production efficiency of enterprises through reasonable production scheduling is crucial. Consequently, ensuring an efficient production rhythm while aiming to reduce total energy consumption has become a research focus for experts and scholars.

Therefore, by optimizing production scheduling, the company can significantly reduce non-cutting time, improve overall production efficiency, and gain an advantage in market competition. Efficient production scheduling helps to reduce various costs in the production process, including time costs, labor costs, material costs, and more, while also improving profit margins. By improving production efficiency and product quality, companies can establish stronger core competitive advantages, and resist market risks and the impact of competitors.

## ACKNOWLEDGEMENT

We sincerely express our deepest gratitude to all individuals and institutions who have contributed to the success of this research project. Special thanks to our families, who not only provided valuable financial support but also served as a source of motivation for us to keep moving forward. At the same time, we are deeply honored and grateful to the companies surveyed, who generously shared their valuable time and insights, laying a solid foundation for in-depth and accurate research.

## DECLARATIONS

### *Conflict of Interest*

There is no conflict of interest in the study.

### *Informed Consent*

Not applicable.

### *Ethics Approval*

Not yet applicable.

## REFERENCES


Chapman, C. D., Saitou, K., & Jakiela, M. J. (2021). Genetic algorithms as an approach to configuration and topology design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 11818, pp. 485-498). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC1993-0338.

Du, Y., Li, J. Q., Chen, X. L., Duan, P. Y., & Pan, Q. K. (2023). Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem. *IEEE Transactions on Emerging Topics in Computational Intelligence, 7*(4), 1036-1050. https://doi.org/10.1109/TETCI.2022.3145706.

Fakhrzad, M. B., Sadeghieh, A., & Emami, L. (2013). *A new multi-objective job shop scheduling with setup times using a hybrid genetic algorithm*. https://sid.ir/paper/589254/en.

Gu, J., Gu, M., Cao, C., & Gu, X. (2010). A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Computers & Operations Research, 37*(5), 927-937. https://doi.org/10.1016/j.cor.2009.07.002.

Jiang, Z., & Le, Z. (2014). Study on multi-objective flexible job-shop scheduling problem considering energy consumption. *Journal of Industrial Engineering and Management (JIEM), 7*(3), 589-604. https://doi.org/10.3926/jiem.1075.

Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32*(1), 1-13. https://doi.org/10.1109/TSMCC.2002.1009117.

Li, J. Q., Pan, Q. K., & Gao, K. Z. (2011). Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. *The International Journal of Advanced Manufacturing Technology, 55*, 1159-1169. https://doi.org/10.1007/s00170-010-3140-2.

Liang, X., Chen, J., Gu, X., & Huang, M. (2021). Improved adaptive non-dominated sorting genetic algorithm with elite strategy for solving multi-objective flexible job-shop scheduling problem. *IEEE Access, 9*, 106352-106362. https://doi.org/10.1109/ACCESS.2021.3098823.

Luo, S., Zhang, L., & Fan, Y. (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Computers & Industrial Engineering, 159*, 107489. https://doi.org/10.1016/j.cie.2021.107489.

Lei, D., Li, M., & Wang, L. (2018). A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold. *IEEE Transactions on Cybernetics, 49*(3), 1097-1109. https://doi.org/10.1109/TCYB.2018.2796119.

Moslehi, G., & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics, 129*(1), 14-22. https://doi.org/10.1016/j.ijpe.2010.08.004.

Momenikorbekandi, A., & Abbod, M. F. (2023). A novel metaheuristic hybrid parthenogenetic algorithm for job shop scheduling problems: Applying an optimization model. *IEEE Access*, *11*, 56027-56045. https://doi.org/10.1109/ACCESS.2023.3278372.

Mantawy, A. H., Abdel-Magid, Y. L., & Selim, S. Z. (1999). Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, *14*(3), 829-836. https://doi.org/10.1109/59.780892.

Piroozfard, H., Wong, K. Y., & Wong, W. P. (2018). Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resources, Conservation and Recycling*, *128*, 267-283. https://doi.org/10.1016/j.resconrec.2016.12.001.

Reijnen, R., Zhang, Y., Bukhsh, Z., & Guzek, M. (2023). Learning to adapt genetic algorithms for multi-objective flexible job shop scheduling problems. In *Proceedings of the Companion Conference on Genetic and Evolutionary* Computation (pp. 315-318). https://doi.org/10.1145/3583133.3590700.

Shi, S., & Xiong, H. (2024). Solving the multi-objective job shop scheduling problems with overtime consideration by an enhanced NSGA-Ⅱ. *Computers & Industrial Engineering*, *190*, 110001. https://doi.org/10.1016/j.cie.2024.110001.

Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, *32*, 12363-12379. https://doi.org/10.1007/s00521-020-04832-8.

Tan, C. J., Hanoun, S., & Lim, C. P. (2015). A multi-objective evolutionary algorithm-based decision support system: A case study on job-shop scheduling in manufacturing. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings* (pp. 170-174). IEEE. https://doi.org/10.1109/SYSCON.2015.7116747.

Tang, D., & Dai, M. (2015). Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem. *Chinese Journal of Mechanical Engineering*, *28*(5), 1048-1055. https://doi.org/10.3901/CJME.2015.0617.082.

Wang, L. (2020). Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity prices. *Knowledge-Based Systems*, *204*, 106177. https://doi.org/10.1016/j.knosys.2020.106177.

Wei, L., He, J., Guo, Z., & Hu, Z. (2023). A multi-objective migrating birds optimization algorithm based on game theory for dynamic flexible job shop scheduling problem. *Expert Systems with Applications*, 227, 120268. https://doi.org/10.1016/j.eswa.2023.120268.

Xing, L. N., Chen, Y. W., & Yang, K. W. (2011). Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems. *Computational Optimization and Applications*, *48*(1), 139-155. https://doi.org/10.1007/s10589-009-9244-7.

Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, *48*(2), 409-425. https://doi.org/10.1016/j.cie.2005.01.018.

Xie, J., Li, X., Gao, L., & Gui, L. (2023). A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems. *Journal of Manufacturing Systems, 71*, 82-94. https://doi.org/10.1016/j.jmsy.2023.09.002.

Yüksel, D., Taşgetiren, M. F., Kandiller, L., & Gao, L. (2020). An energy-efficient bi-objective no-wait permutation flow shop scheduling problem to minimize total tardiness and total energy consumption. *Computers & Industrial Engineering, 145,* 106431. https://doi.org/10.1016/j.cie.2020.106431.

Yu, F., Lu, C., Zhou, J., Yin, L., & Wang, K. (2024). A knowledge-guided bi-population evolutionary algorithm for energy-efficient scheduling of distributed flexible job shop problem. *Engineering Applications of Artificial Intelligence, 128*, 107458. https://doi.org/10.1016/j.engappai.2023.107458.

Yin, G., Ma, M., Jia, P., & Ma, X. (2024). Parameter optimization of friction pendulum bearings based on the adaptive genetic algorithm considering the overall evolutionary status. *Buildings*, *14*(2), 435.

Zhang, G., Hu, Y., Sun, J., & Zhang, W. (2020). An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm and Evolutionary Computation, 54*, 100664. https://doi.org/10.1016/j.swevo.2020.100664.

Zhang, X., Zhang, H., & Yao, J. (2020). Multi-objective optimization of integrated process planning and scheduling considering energy savings. *Energies*, *13*(23), 6181. https://doi.org/10.3390/en13236181.

Zuo, Y., Wang, P., & Li, M. (2023). A population diversity-based artificial bee colony algorithm for assembly hybrid flow shop scheduling with energy consumption. *Applied Sciences*, *13*(19), 10903. https://doi.org/10.3390/app131910903.

**Author's Biography**

Lei Chen received a bachelor's degree from Nanchang Hangkong University in 2010 and a master's degree from the Guangdong University of Technology in 2013. He is currently a lecturer at Jiangxi College of Applied Technology and is pursuing a doctorate in Information Technology at the University of the East, Manila in the Philippines. His research interest includes mechatronics, electronics, and information technology.

Dr. Joan P. Lazaro is a full-time professor at the College of Engineering, Computer Engineering Department, and a special lecturer of IT programs at the Graduate School at the University of the East. He is a graduate of Doctor of Information Technology and Master of Engineering Science from the University of the East Manila Graduate School and a Bachelor of Science in Computer Engineering from the University of the East Caloocan. Among the different certifications he earned are the following: Professional Computer Engineer, Fortinet's Network Security Expert Certification – NSE 1 and 2 Network Security Associate, Certified Microsoft Innovative Educator Program, and National Certificate II in Mechatronics Servicing. His research interests include software development, network security, and engineering sciences.