



Parallel batch processor machine scheduling using multi population strength pareto evolutionary algorithm 2

Ferdinan Rinaldo Tampubolon ¹, Sinta Marito Siagian ², Samaria Chrisna HS ³, Samaria Chrisna HS ³, Risha Devita⁴ Anna Angela Sitinjak ⁵

^{1,2,3,4} Teknik Elektro, Teknik Elektronika, Politeknik Negeri Medan, Indonesia

⁵ Teknik Mesin, Politeknik Teknologi Kimia Industri, Indonesia

Article Info

Article history:

Received Mar 28, 2025

Revised May 29, 2025

Accepted Jun 25, 2025

Keywords:

Batch;
Flexible;
Job Shop;
Multi-Population;
SPEA2;

ABSTRACT

Flexible Job Shop with Parallel Batch Processor (FJSP-BP) has been studied by several researchers, but the objective function has generally been limited to minimizing makespan. With the growing awareness towards environmental issues, it has become increasingly important to incorporate objective function that address environmental impacts, such as energy consumption. This research using two objective functions in FJSP-BP scheduling, namely makespan and energy consumption. MultiPopulation Strength Pareto Evolutionary Algorithm 2 (MP-SPEA2) will be used to solve that multiobjective scheduling. The concept of Multi population on classical SPEA2 are conducted by applying several sub population that run in parallel so that the optimization process can avoid local optima. The results of the research show that MP-SPEA2 provides more optimal results compared to classical SPEA2 and benchmarks from previous research. Therefore, it can be used as an alternative to solve similar cases in the industrial sector to produce machine scheduling that are not only profitable but also eco-friendly.

This is an open access article under the CC BY-NC license.



Corresponding Author:

Ferdinan Rinaldo Tampubolon,
Teknik Elektro/Teknik Elektronika,
Politeknik Negeri Medan,
Jl. Almamater No.1 Padang Bulan Medan, Sumatera Utara, 20155, Indonesia
Email: ferdinantampubolon@polmed.ac.id

1. INTRODUCTION

Production scheduling is a process of organizing, controlling and improving work efficiency in a factory. This activity distributes jobs to each machine within a certain time period to optimize a specific objective function [1]. Production scheduling can be divided into several categories depending on the production flow, which are flow shop, job shop, and open shop [2]. In Flow Shop scheduling, a set of machines is arranged in series and every job undergoes identical processing sequence on each machine [3]. In Job Shop scheduling, there are jobs that have different machine processing routes compared to other jobs. Job shop scheduling is often used in manufacturing environments that produce customized products [4]. Research on job shop scheduling has evolved with the introduction of flexibility, where a job can be processed by more than one machine, which differs from the traditional job shop where each job can only be processed by one specific machine. This scheduling is called Flexible Job Shop Scheduling Problem (FJSSP) [5].

With the increasing competition in the Industrial sector, companies are required to provide more optimized services in terms of production speed, one of which is by increasing machine utility. This can be done by using batch scheduling. In conventional scheduling, the machine can only process one job, but in batch scheduling the machine can process a set of jobs according to the capacity of the

machine. The set of jobs will be processed with the same start time and completion time. Batch scheduling is divided into two, namely serial batch, where each individual job in the batch is processed sequentially (in series). Next, parallel batch, where each job is completed simultaneously (parallel) [6]. Batch scheduling has been widely applied in industries such as aircraft, steel processing, glass manufacturing and semiconductor industry [7]. Although it increases the machine utility, the total capacity of the batch will be crucial, if there are too many jobs assigned to the batch, the batch machine will become a bottleneck in the production flow because the processing time of a job is the collective completion time of all the jobs in the batch, delaying the processing of the job for the next machine, while if there are too few jobs assigned to the batch machine will reduce the machine utility [8]. Therefore, an optimal batch scheduling method is required.

The first research related to Flexible Job Shop with parallel batch processing was conducted by [9] using an exact scheduling method, Mixed Integer Linear Programming (MILP) but this method does not run optimally for the case of a large number of jobs and machines, research using heuristic methods is more appropriate for cases with a larger scale, some researchers use heuristic methods, such as, hybrid Particle Swarm Optimization (PSO) methods [10]. Hybrid Large Neighborhood Search (LNS) [11] and most recently hybrid multi population genetic algorithm [12]. All of these researches use a single objective function, which is to minimize the overall job completion time (makespan).

Energy-based scheduling is a topic that is often discussed with the rising cost of energy [13]. Some studies use energy consumption objective function to involve sustainability aspects in production planning [14] as performed by [15] and [16] in case of FJSSP. Research related to the energy consumption objective function has reached 15 percent of all production scheduling articles in 2021, under the makespan and tardiness objective functions. The use of more than one objective function (multiobjective) in certain situations will be more relevant in the industrial world [17].

This research focuses on two objective functions, namely makespan and energy consumption in Flexible Job Shop Scheduling with Parallel Batch Processors. The makespan objective is related to operational aspects, it aims to schedule jobs in a way that minimizes the total completion time. On the other hand, energy consumption is linked to environmental concerns. Therefore, combining these two objective functions results in a scheduling approach that not only optimizes operational efficiency but also addresses environmental sustainability. Generally, multi-objective optimization cannot find a single minimum solution for each objective function simultaneously, because there will be a trade-off [18] among one objective function and another. For example, scheduling with a makespan objective function will allocate jobs to the machine with the fastest processing time without considering the energy consumption of the machine. Meanwhile, scheduling with an energy consumption objective function will focus on allocating jobs to machines with the least power usage which may sacrifice job completion time. Thus, multi-objective optimization will search a solution that is good enough and not dominated by other solutions, this solution is called pareto optimal. Methods using pareto that are quite often used are Nondominated Sorting Genetic Algorithm (NSGA-II) [19] and Strength Pareto Evolutionary Algorithm SPEA2 [20]. The SPEA2 algorithm is a type of evolutionary algorithm that imitate the concept of evolution in living organisms, such as selection, crossover, and mutation. The SPEA2 algorithm uses the concept of "strength" to determine the quality of a solution. A solution is evaluated based on the number of other solutions it dominates; solutions that dominate others are stored in an archive to ensure they are not lost during crossover or mutation.

Research [21] shows that the SPEA2 method has better performance and more diverse solutions compared to NSGA-II for objective functions with no more than two objectives. However, these Pareto-based algorithms are prone to getting stuck in local optima because they use only a single population in each evolutionary step. In multi-population, the population will be divided into several subpopulations, each of these subpopulations will run in parallel with different parameters or types of crossover and mutation, Therefore it will be able to explore more varied solutions that may not be found in classic SPEA2. Research related to multi-population was conducted by [22] where the results showed the Multi-Population NSGA-II improved the quality of solutions compared to the NSGA-II algorithm.

This research uses Multi population SPEA2 to optimize Flexible Job Shop scheduling with parallel batch processors in terms of time and energy. There are two novelties in this research, the first is related to the use of more than one objective function, namely makespan and energy consumption in Flexible Job Shop Scheduling with Parallel Batch Processing. The second is the use of a modification of the SPEA2 algorithm, namely Multi population SPEA2 which is relatively new in solving these scheduling. The results of this research are expected to be useful in industrial sectors that use similar scheduling flows such as semiconductors, to produce machine scheduling that is not only profitable but also eco-friendly.

2. RESEARCH METHOD

2.1. Flexible Job Shop Batch Processing Machine

The mathematical model for Flexible Job Shop Scheduling with Parallel Batch Processors is as follows:

Objective Function

$$\min (\max(E_{ki})) \quad (1)$$

Constraints

$$Z_1 - Z_2 > 0 \quad (2)$$

$$Y_{ij} + Y_{ji} = 1 \forall k, i, j \quad (3)$$

$$S_{ki} \geq 0, E_{ki} > 0, \forall i, j \quad (4)$$

$$Z_3 \leq C_m \quad (5)$$

$$E_{ki} = S_{ki} + T_{ki}, \forall k, i \quad (6)$$

$$\sum_i X_{kim} = 1, \forall k, i, m \quad (7)$$

$$S_{ki} = S_{ij}, E_{kj} = E_{lj}, \forall i, j \in B_{mb} \quad (8)$$

$$P_{kiam} + P_{kibm} + P_{kian} + P_{kbn} = 1, \forall a \neq b, m \neq n, k, i \quad (9)$$

Var	Definition	Var	Definition
i, j	Operation index	B_{nb}	The b-th batch on the n-th parallel batch machine
k, l	job index	U	set of operation proceeded by batch processing machine
m, n	machine index	M_{ki}	set of machine for proceeding the O_{ki} operation
b	batch index	D_{ki}	capacity demand for each set operation in U
O_{ki}	The i-th operation of job-k	Y_{kijl}	equals to 1 if O_{ki} precedes operation O_{kj} and 0 otherwise
S_{ki}	Starting time of O_{ki}	Z_2	auxiliary variable, if E_{ki} multiplied by Y_{kij} ,
E_{ki}	completion time of O_{ki}	Z_3	Auxiliary Variable if D_{ki} multiplied by P_{kibn}
T_{kim}	Processing time of operation O_{ki} on machine m	X_{kim}	$\begin{cases} 1, & O_{ki} \in m \\ 0, & \text{otherwise} \end{cases}$
Z_1	auxiliary variable if S_{ki} multiplied by Y_{kij}	P_{kibm}	$\begin{cases} 1, & \text{if } O_{ki} \text{ processed on batch } b \text{ in } n - \text{th parallel} \\ 0, & \text{otherwise} \end{cases}$

Equation (1) show final completion time that is the largest processing time. Equation (2) and (3) ensure the order of the completion of operations and assuring that each operation is processed once. Equation (4) ensures the starting time and completion time are positive. Equation (5) shows the capacity limitation of the batch machine. Equation (6) shows the definition of operation completion time. Equation (7) ensures each operation can only be processed by one machine, Equation (8) shows the limitation of batch machine processing time, Equation (9) shows each operation can only be in a single batch [12].

2.2. Energy Consumption

Energy consumption in Flexible Job Shop scheduling is obtained by calculating the total energy consumption of machines when they are working and idle. The energy consumption of a working

machine is the total energy consumption of the machine when processing an operation O_{ki} multiplied by the completion time of the operation E_{ki} . Meanwhile, the energy consumption when idle is the energy consumption of the machine when it is running but not processing an operation. Energy consumption on a parallel batch processor machine for n-jobs are as follow: The processing time batch- b is determined by the completion time of the longest job in the batch. Let S_b , and C_b denotes start time and completion time of the batch. The machine cannot be shut down until the job processing is completed. The electricity consumption during processing time is denoted as PP and the electricity consumption during idle time is denoted as IP . The total energy consumption is as follows

$$EC = \sum_{b=0}^{n-1} \sum_{t=S_b+1}^{C_b} IP_{bt} + \sum_{b=1}^n \sum_{t=S_b+1}^{C_b} PP_{bt} \quad (10)$$

Total Energy Consumption is in form of electricity consumption during working and idle [23].

2.3. Multi-Population SPEA-2 Algorithm

Strength Pareto Evolutionary Algorithm 2 is an algorithm based on evolutionary principles used to solve multiobjective problems. This algorithm uses the concept of “strength”, which ranks solutions based on strength, in other words, the number of dominated solutions. The fundamental procedure of SPEA2 is as follows

Step 1: Initialize individuals for each population, determine the variable P_0 , and the archive variable $\bar{P}_0$ with the number of iterations $t = 0$. For illustration, the example individual as shown in the figure 1

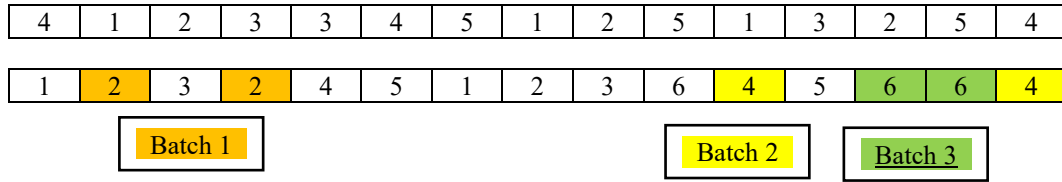


Figure 1. Individual

The individuals consist of two chromosomes: job order and machine assignment, The chromosome consists of several genes, the number 4 in the first gene and the first chromosome indicates that operation O_{41} is processed first then followed by operation O_{11} and so forth. The second chromosome shows the machine assignment, which means O_{41} is processed by machine 1. Operations O_{11} and O_{31} are processed simultaneously in one batch 1 by machine 2.

Step 2: Determine the Fitness Value $F(i)$ for each individual in population

Step 3: Selection Process. Insert non dominated solution P_t and $\bar{P}_{t+1}$ and if $\bar{P}_{t+1}$ greater than $\bar{N}$, which is the total population, perform the pruning process using the adaptation degree.

$$S(i) = \{j | j \in P_t + \bar{P}_t \cap j < i\}$$

$$R(i) = \sum_{j \in P_t + \bar{P}_t, i < j} S(j)$$

where $S(i)$ denotes amount of individuals that are dominated by individual i . $R(i)$ denotes the adaptation value which contains the amount of individuals i that are dominated by j . k -neighborhood is used to estimate the density. The distance between solutions in the population is expressed in the order σ_i^k where $K = \sqrt{N + \bar{N}}$ to ensure the value is within the interval (0,1)

Distance of solutions in the population $D(i)$.

$$D(i) = \frac{1}{\sigma_i^k}$$

Thus, the fitness value is

$$F(i) = R(i) + D(i)$$

Step 4: When $t > T$ or any other condition is met, the new non-dominant solution $\bar{P}_{t+1}$ is displayed

Step 5: Perform tournament selection on $\bar{P}_{t+1}$ to be added on *mating pool*.

Step 6: Select solutions from mating pool to undergo crossover and mutation process and return to the second step [20].

Figure 1 illustrates two chromosomes that will undergo crossovers process, the crossovers between these chromosomes will produce two offspring as shown in Figure 3

4	1	2	3	3	4	5	1	2	5	1	3	2	5	4
3	1	2	4	5	1	4	3	2	5	4	1	3	2	5

Figure 2. Parent Chromosome

3	3	4	5	1	2	1	2	4	2	5	4	1	3	5
4	1	2	3	4	5	1	3	2	5	1	3	2	5	3

Figure 3. Offspring Chromosome

The mutation process is performed by exchanging several genes from the initial chromosome. This gene exchange will produce a new solution. One form of mutation is insertion, as shown in Figure 4.

4	1	2	3	3	4	5	1	2	5	1	3	2	5	4
3	3	4	5	1	2	5	1	3	2	4	1	2	5	4

Figure 4. Mutation Process

In the concept of multi-population, the population in SPEA2 is not single, there are several populations that will run simultaneously. Each individual in the population at a certain condition can move (migrate) to another population, this is necessary to increase the variation of the solution so it doesn't become trapped in the local optimum [24].

The Multi Population process as shown in Fig 5 begins by generating a set of individuals or chromosomes to form a population. This population will be partitioned into several subpopulations. Each subpopulation will undergo a selection process. Individuals in the subpopulation that have the best fitness value will be kept in the archive set, which is a collection of non-dominated solutions, while other solutions will enter the mating pool to be selected as parents which will then produce offspring through the crossover. Mutation will also be performed on the selected solutions based on the mutation probability. The next stage is integration process where non-dominated solutions will be merged with other solutions. Each subpopulation can exchange its individual at certain iterations to explore the search space. If the maximum number of iterations has not been reached, the process returns to the selection step, otherwise it is terminated.

2.4. Methodology

Nine cases used in this study are the benchmark instances MK1–MK9 from Brandimarte [25], where each case consists of various number of jobs, operations and machines, for example for MK9 consists of 20 jobs, 10 machines, a range of 10–15 operations for each job. Research [12] modified the case by adding parallel batch processing machine in each case. Here an example modified case for MK1 and MK2: 1-3, 5-4, 6-3, 7-5, 8-6, 9-5. For the MK1 case, 1-3 denotes O_{13} , and 5-4 denotes O_{56} and so forth. These six operations will be performed by the batch processing machine simultaneously. The batch processing machine in this particular instance has a capacity of 10, which means it can process 10 operations simultaneously.

Since based on the author's observation, there has not been any research related to the FJSSP case with batch processing machines that considers energy consumption Hence in this research random numbers are generated regarding energy consumption $t=0.01\sim0.03$ for non-batch machines in idle state, and $t=0.1\sim0.3$ when working, while for batch machines energy consumption $t=0.03\sim0.05$, in idle state and $t=0.7\sim0.9$ when working.



Figure 5. Flowchart Multi-Population SPEA2 Algorithm

The SPEA2 algorithm is used to solve the FJSP case with a parallel batch processor for MK₁-MK₉ data and compare the results with benchmarks from previous research. Since both research only use one objective function, the SPEA2 algorithm will be considered successful if it has a makespan value that is close to or even smaller than the benchmark from previous studies. Furthermore, the multi-population algorithm will be compared with the SPEA2 algorithm on the same cases. Each algorithm is executed five times, and the set of non-dominated solutions will be presented.

The parameters used in the SPEA2 algorithm are as follows: population size ranges from 100 to 200, maximum iterations ranges from 100 to 300, crossover probability ranges from 0.5 to 0.8, and mutation probability ranges from 0.4 to 0.7.

The parameters used in MP-SPEA2 algorithm, four subpopulations were used in this study. Each subpopulation consisted of 25 to 50 chromosomes and performed a total of 100 to 300 iterations. Migration, or information exchange, occurred every five iterations, involving the exchange of some of the best and worst individuals between subpopulations. In addition, random exchanges of individuals were conducted to avoid premature convergence. Each subpopulation used different crossover and mutation probabilities, such as 0.8 and 0.2; 0.5 and 0.5; 0.8 and 0.7; 0.2 and 0.8.

3. RESULTS AND DISCUSSIONS

3.1. Result

The results using SPEA2 with makespan and energy consumption as shown in Table 1:

Table 1. Makespan Comparison

Case	SPEA2		Makespan [25]	Makespan [12]
	Makespan	Energy		
MK ₁	36	34,09	42-65	40-58
	37	32,37		
MK ₂	35	25,88	32-48	43-45
MK ₃	136	134,22	211-506	247-359
	137	133,59		
	141	132,72		
MK ₄	92	85,38	81-121	74-99
	100	83,44		
MK ₅	234	154,62	186-291	164-207
MK ₆	85	91,26	86-128	111-187
MK ₇	123	117,58	157-262	111-187
	119	118,45		
	116	118,47		
MK ₈	347	454,71	523-1042	479-586
MK ₉	363	472,5	369-723	360-466
	383	468,2		
MK ₁₀	351	389,8	296-536	326-467
	364	383,98		

Table 1 shows the results obtained by the SPEA2 algorithm. The SPEA2 algorithm is an algorithm designed to solve cases with multiple objective functions, and in this case the objective functions are makespan and energy consumption. For MK₁ case, this algorithm produces a non-dominated solution, first solution with a makespan of 36 and energy consumption of 34.09 and second solution with a makespan of 37 and energy consumption of 32.37. The lower the makespan and energy consumption, the more desirable that solution becomes. The first solution produces a lower makespan than the second solution, but the second solution produces a smaller energy consumption than the first solution so that the two solutions do not dominate each other (equally strong). The makespan value represents the total completion time for all jobs. For example, if the makespan is 36, it means that all jobs are completed by the machines in 36, measured in time units. Meanwhile, the energy consumption value indicates the total amount of energy used by the machines to complete all jobs. For instance, a value of 34.09 indicates that the total energy consumed is 34.09, measured in energy units.

Meanwhile, for the comparison of makespan results (benchmark), previous research [25]. obtained makespan with several methods and had the lowest makespan value of 42. When compared to some previous studies, even though this research uses more than one objective function, namely makespan and energy consumption. The SPEA2 algorithm produces makespan values that are close to and even better for some cases compared to existing benchmarks.

As opposed to the standard SPEA2 Algorithm that uses only one population and evolves in several iterations, Multi Population SPEA2 divides the population into several subpopulations, where each of these subpopulations will undergo an evolutionary process in parallel and independently. Each of these subpopulations under certain conditions will exchange information, in this case the exchange of chromosomes so that the search for solutions will be more varied and will accelerate convergence. The results obtained by using the SPEA2 Multi population algorithm are as shown in Table 2

Table 2. Comparison Using SPEA-2 and Multi Population SPEA2

Case	SPEA2		Multi Population SPEA2	
	Makespan	Energy	Makespan	Energy
MK ₁	36	34,09	32	32,39
	37	32,37	34	32,22
MK ₂	35	25,88	33	26,43
MK ₃	136	134,22	136	130,18
	137	133,59	136	131,11
	141	132,72	137	128,86

Case	SPEA2		Multi Population SPEA2	
	Makespan	Energy	Makespan	Energy
MK4	92	85,38	86	89,2
	100	83,44		
MK5	234	154,62	232	159,7
MK6	85	91,26	82	94,26
MK7	123	117,58	104	116,53
	119	118,45		
	116	118,47		
MK8	347	454,71	344	452,75
MK9	363	472,5	332	437,73
	383	468,2		
MK10	351	389,8	325	382,91
	364	383,98		

Based on testing for 10 FJSSP cases with batch processing machines the Multi population SPEA2 (MP-SPEA2) algorithm excels in several cases except in cases, MK2, MK4, MK5, MK6, in that cases although the (MP-SPEA2) algorithm produces lower makespan, but the SPEA2 algorithm has lower energy consumption.

3.2. Discussion

Despite having two objective functions, the multi population SPEA2 algorithm produces makespan values that are comparable to or even better than previous studies which uses only one objective function, except for the MK4 and MK5 cases. The proposed algorithm produces better quality solutions in 6 cases, MK1, MK3, MK7, MK8, MK9, MK10 compared to the SPEA2 algorithm, although in four other cases MP-SPEA2 is not superior, SPEA2 does not have a more dominant solution than MP-SPEA2. Distribution of MK4 case solutions generated by these two algorithms. Figure 6 shows some solution using the SPEA2 algorithm for several iterations. The vertical axis represents the makespan, while the horizontal axis shows the results for the energy consumption. Figure 6(a) presents the solution at the 50th iteration, where the distribution of each solution is still quite wide, and both objective function that is makespan and energy consumption values remain relatively high. In Figure 6(b), the solution distribution is still wide, but the values of both objective functions begin to decrease. In Figures 6(c) and 6(d), the solutions exhibit decreasing objective function values and a distribution that becomes more concentrated toward the optimal area. A clear trade-off is observed, where solutions with lower makespan values tend to have higher energy consumption, whereas solutions with lower energy consumption typically have higher makespan values.

For the MP-SPEA2 algorithm, the distribution of solutions shown in Figure 7. In Figure 7(a), at the 5th iteration, the solutions still exhibit a wide distribution, however, the range of makespan and energy consumption values is smaller compared to SPEA2. In Figure 7(b), the solutions have already toward converged, leaving only two optimal non-dominated solutions. The use of multiple populations running in parallel in MP-SPEA2 allows this algorithm to have different crossover, mutation parameters, and archive sets for each subpopulation, in contrast to the standard SPEA2, which uses only a single set of parameters and an archive set. In addition, migration of solutions between subpopulations enables MP-SPEA2 to explore a wider solution space compared to the standard SPEA2 algorithm. This increases the likelihood of discovering solutions with better trade-offs and accelerates the convergence process.

In industries that implement FJSP scheduling with parallel batch processors, there are various types of jobs that must be processed by a limited number of machines, each with different processing times and energy consumption levels. The MP-SPEA2 algorithm can be applied to determine the sequence of jobs, assign machines for each job, and decide which jobs should be grouped into batches, aiming to achieve optimal scheduling in terms of both completion time and energy consumption. By minimizing energy consumption, companies can reduce energy costs, lower pollution, and enhance their reputation, all of which are closely related to corporate sustainability

4. CONCLUSION

The proposed Multi population SPEA2 algorithm demonstrates optimal performance in solving the Flexible Job Shop Scheduling Problem (FJSSP) involving batch processors to simultaneously minimize two objective functions, namely makespan and energy consumption. By dividing the population in SPEA2 into several subpopulations, Multi Population SPEA2 improves the performance of the classical SPEA2 algorithm. This algorithm is expected to serve as an alternative solution for similar cases in industry, enabling production processes that are not only time-efficient but also environmentally friendly. The multi-population process can also be applied to existing metaheuristic algorithms to accelerate convergence. However, this study has limitations since the data used is static and deterministic. This study does not take into account uncertain processing times, the arrival of new jobs, machine breakdowns, or other events that require real-time rescheduling. Under such conditions, the use of metaheuristic algorithms would require considerable computational time, since the algorithm would need to be run multiple times for each rescheduling event. Therefore, combining reinforcement learning with metaheuristic concepts can be utilized to address scheduling processes involving dynamic and probabilistic data for future research.

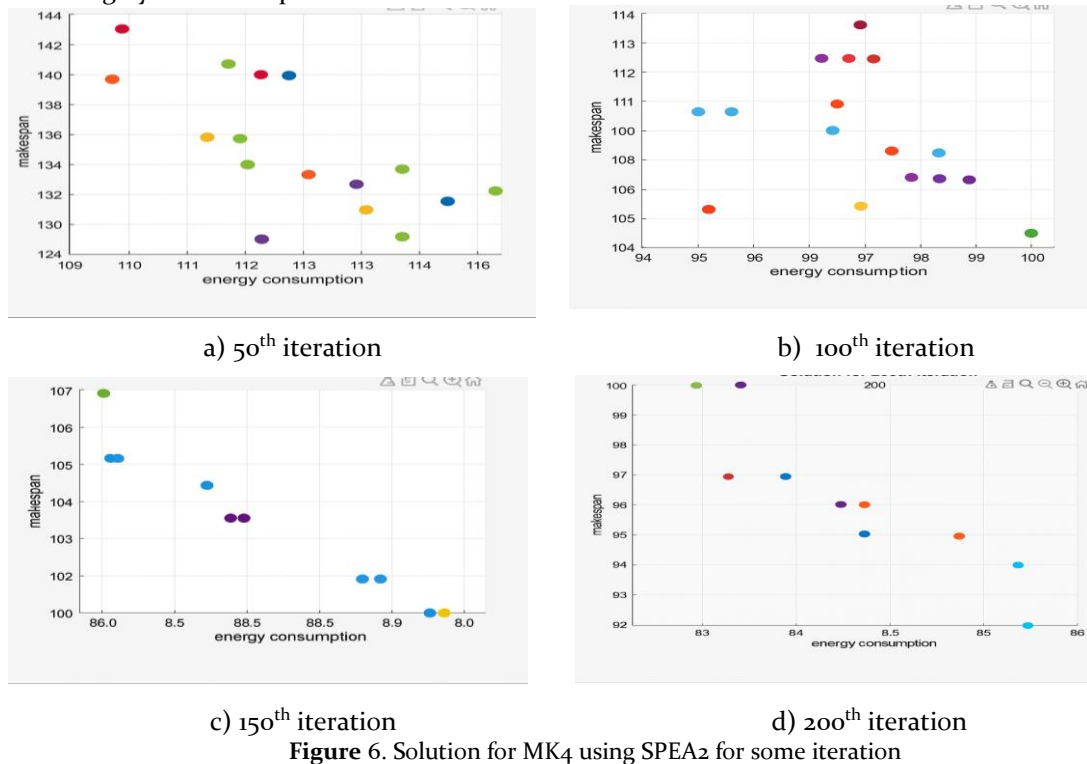


Figure 6. Solution for MK4 using SPEA2 for some iteration

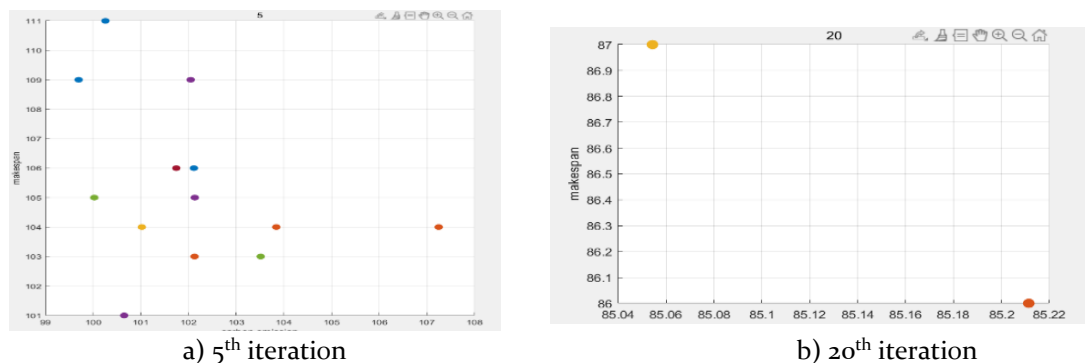


Figure 7. Solution for MK4 using MP-SPEA2 for some iteration

REFERENCES

- [1] M. L. R. Varela, G. D. Putnik, V. K. Manupati, G. Rajyalakshmi, J. Trojanowska, and J. Machado, "Integrated process planning and scheduling in networked manufacturing systems for I4.0: a review and framework proposal," *Wirel. Networks*, vol. 27, no. 3, pp. 1587–1599, 2021, doi: 10.1007/s11276-019-02082-8.
- [2] V. T'Kindt and J. C. Billaut, *Multicriteria scheduling: Theory, models and algorithms*, Second. Springer, 2006. doi: 10.1007/b106275.
- [3] M. Karim, O. Hamlaoui, and H. Hadda, "Regular articles A simulated annealing metaheuristic approach to hybrid flow shop scheduling problem," *Adv. Ind. Manuf. Eng.*, vol. 9, no. December 2023, p. 100144, 2024, doi: 10.1016/j.aime.2024.100144.
- [4] G. Da Col and E. C. Teppan, "Industrial-size job shop scheduling with constraint programming," *Oper. Res. Perspect.*, vol. 9, no. March, p. 100249, 2022, doi: 10.1016/j.orp.2022.100249.
- [5] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, "The flexible job shop scheduling problem: A review," *Eur. J. Oper. Res.*, vol. 314, no. 2, pp. 409–432, 2024, doi: 10.1016/j.ejor.2023.05.017.
- [6] O. Ozturk, "When serial batch scheduling involves parallel batching decisions: A branch and price scheme," *Comput. Oper. Res.*, vol. 137, no. July 2020, p. 105514, 2022, doi: 10.1016/j.cor.2021.105514.
- [7] J. Xu, J. Q. Wang, and Z. Liu, "Parallel batch scheduling: Impact of increasing machine capacity," *Omega (United Kingdom)*, vol. 108, p. 102567, 2022, doi: 10.1016/j.omega.2021.102567.
- [8] Y. Li, A. Wang, and S. Zhang, "A Batch Scheduling Technique of Flexible Job-Shop Based on Improved Genetic Algorithm," 2022 *IEEE Int. Conf. Mechatronics Autom. ICMA 2022*, pp. 1463–1467, 2022, doi: 10.1109/ICMA54519.2022.9856332.
- [9] A. Ham, "Flexible job shop scheduling problem for parallel batch processing machine with compatible job families," *Appl. Math. Model.*, vol. 45, pp. 551–562, 2017, doi: 10.1016/j.apm.2016.12.034.
- [10] L. Song, C. Liu, and H. Shi, "Discrete particle swarm algorithm with Q-Learning for solving flexible job shop scheduling problem with parallel batch processing machine," *J. Phys. Conf. Ser.*, vol. 2303, no. 1, pp. 1–11, 2022, doi: 10.1088/1742-6596/2303/1/012022.
- [11] B. Ji, S. Zhang, S. S. Yu, and B. Zhang, "Mathematical Modeling and A Novel Heuristic Method for Flexible Job-Shop Batch Scheduling Problem with Incompatible Jobs," *Sustain.*, vol. 15, no. 3, 2023, doi: 10.3390/su15031954.
- [12] L. Xue, S. Zhao, A. Mahmoudi, and M. R. Feylizadeh, "Flexible job-shop scheduling problem with parallel batch machines based on an enhanced multi-population genetic algorithm," *Complex Intell. Syst.*, vol. 10, no. 3, pp. 4083–4101, 2024, doi: 10.1007/s40747-024-01374-7.
- [13] O. A. Olanrewaju, F. Luiz, and P. Krykhtine, "Minimum-Energy Scheduling of Flexible Job-Shop Through Optimization and Comprehensive Heuristic," 2024.
- [14] M. Danishvar, S. Danishvar, E. Katsou, S. A. Mansouri, and A. Mousavi, "Energy-Aware Flowshop Scheduling: A Case for AI-Driven Sustainable Manufacturing," *IEEE Access*, vol. 9, pp. 141678–141692, 2021, doi: 10.1109/ACCESS.2021.3120126.
- [15] H. Terbrack and T. Claus, "Computers & Industrial Engineering The generalized energy-aware flexible job shop scheduling model : A constraint programming approach," *Comput. Ind. Eng.*, vol. 204, no. March, p. 111065, 2025, doi: 10.1016/j.cie.2025.111065.
- [16] S. Christian, B. Daniela, and G. Guido, "A memetic NSGA - II for the multi - objective flexible job shop scheduling problem with real - time energy tariffs," *Flex. Serv. Manuf. J.*, vol. 36, no. 4, pp. 1530–1570, 2024, doi: 10.1007/s10696-023-09517-7.
- [17] D. Alemão, A. D. Rocha, and J. Barata, "Smart manufacturing scheduling approaches—systematic review and future directions," *Appl. Sci.*, vol. 11, no. 5, pp. 1–20, 2021, doi: 10.3390/app11052186.
- [18] P. K. Shukla, C. Hirsch, and H. Schmeck, "Towards a Deeper Understanding of Trade-offs Using Multi-objective Evolutionary Algorithms," no. Mcdm, pp. 396–397, 2012.
- [19] F. Luan, H. Zhao, S. Qiang, Y. He, and B. Tang, "Sustainable Computing : Informatics and Systems Enhanced NSGA-II for multi-objective energy-saving flexible job shop scheduling," vol. 39, no. April 2022, pp. 0–2, 2023.
- [20] S. Larraín, L. Pradenas, I. Pulkkinen, and F. Santander, "Multiobjective optimization of a continuous kraft pulp digester using SPEA2," *Comput. Chem. Eng.*, vol. 143, 2020, doi: 10.1016/j.compchemeng.2020.107086.
- [21] I. Huseyinov and A. Bayrakdar, "Performance Evaluation of NSGA-III and SPEA2 in Solving a Multi-Objective Single-Period Multi-Item Inventory Problem," *UBMK 2019 - Proceedings, 4th Int. Conf. Comput. Sci. Eng.*, no. 4, pp. 531–535, 2019, doi: 10.1109/UBMK.2019.8907139.
- [22] Z. Zhao, B. Liu, C. Zhang, and H. Liu, "An improved adaptive NSGA-II with multi-population algorithm," pp. 569–580, 2019.

- [23] S. Zhou, M. Jin, and N. Du, "Energy-efficient scheduling of a single batch processing machine with dynamic job arrival times," *Energy*, vol. 209, 2020, doi: 10.1016/j.energy.2020.118420.
- [24] D. Li, Q. Hou, M. Zhao, and Z. Wu, "Reliable Task Planning of Networked Devices as a Multi-Objective Problem Using NSGA-II and Reinforcement Learning," *IEEE Access*, vol. 10, pp. 6684–6695, 2022, doi: 10.1109/ACCESS.2022.3141912.
- [25] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Ann. Oper. Res.*, vol. 41, no. 3, pp. 157–183, 1993, doi: 10.1007/BF02023073.