International Journal of Industrial Engineering Computations 15 (2024) 871–886

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Energy-efficient scheduling for a flexible job shop problem considering rework processes and new job arrival

Emrah Albayrak[a*] and Semih Önüt[a]

[a]*Yıldız Teknik University, Department of Industrial Engineering, Istanbul, Turkey*

C H R O N I C L E

Article history:
Received March 17 2024
Received in Revised Format June 20 2024
Accepted July 27 2024
Available online July 27 2024

Keywords:
Energy-efficient
Enhanced NSGA II
Rescheduling
Rework processes
Multi-objective optimization
Flexible job shop scheduling

A B S T R A C T

Sustainable production is not limited to environmental concerns only; It also provides economic benefits for businesses. Businesses that adopt sustainability principles can gain advantages in matters such as cost savings, competitive advantage, risk management, legal compliance and corporate reputation. Therefore, sustainability is no longer an option but a strategic imperative for businesses. For this reason, studies on energy-sensitive scheduling have started to increase recently. Another important factor in sustainable manufacturing is the reduction of scrap. Rework operations are required to reduce scrap. In this study, the multi-objective flexible job shop scheduling problem (MO-FJSP) that considers energy efficiency is discussed. The created model aims to minimize the energy consumption, total machine workload and makespan. In this study, new job arrivals are considered as dynamic events. Another dynamic event added to the model is the addition of rework processes between operations to reduce the scrap rate when a scrap decision is made during the production stages. The enhanced NSGA II algorithm was applied to solve this problem. The enhanced NSGA II algorithm was applied to test instances and its performance was compared using some of the multi-objective performance indicators. These experimental results prove the effectiveness of the proposed solution method.

1. Introduction

The current state of the ecosystem is negatively impacted by increased global industrialization, population growth, development of new products, high production, and excessive consumption, all of which are factors in economic development. Sustainable production is an important strategy, both environmentally and economically, and plays a critical role in the long-term success of businesses (Xu et al., 2016). Therefore, production companies try to reduce energy consumption and minimize their environmental impact by adopting sustainability principles. This is a strategy that aims to both create a green business and maintain economic stability in the long term.

Scheduling is an optimization problem. The main purpose of scheduling can be considered as ensuring sustainable production, using production resources most efficiently, responding to customer demands immediately, avoiding delays in product deliveries, reducing inventory costs and overtime work. Today, with increasing energy costs, energy shortage, global warming and environmental pollution, studies to reduce energy consumption in scheduling problems have started to increase.

In many production processes, some of the products produced may be defective due to non-perfect technology or human mistakes. Instead of separating defective products for scrap, they are reworked to recover material and add value. These rework activities are also supported by increased environmental awareness and costs. Rework processes make scheduling problems even harder. An extension of conventional job shop scheduling problem, the flexible job shop scheduling problem (FJSP), has been proven to be NP-hard (Li & Gao, 2016).

* Corresponding author
E-mail emrah.albayrak@std.yildiz.edu.tr (E. Albayrak)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2024.7.004

In manufacturing systems, the production process is generally not static. Manufacturing processes are often interrupted by dynamic events during production. Some dynamic events are new job arrivals and rework processes. In manufacturing systems where raw material costs are high, additional operations are added to existing operations to reduce scrap costs. These additional operations are called rework processes. The produced product is becoming suitable for customer requirements through rework processes.

The model created in this study aims to minimize energy consumption, total machine workload and makespan. Additionally, the created model includes dynamic events based on new job arrivals and rework processes. The main purpose of the model is not to shorten the production time of jobs, as is usually done, but also to balance it from a sustainable perspective, to make green manufacturing and to provide a competitive advantage by reducing scrap costs.

The rest of the research is organized as follows. In section 2, a literature search was conducted. In Section 3, the mathematical model was defined and discussed. The enhanced NSGA II algorithm was designed and described in Section 4. The comparison experiment of enhanced NSGA II and some other multi-objective algorithms and discussion for the results of the experiment were presented in Section 5. Finally, Section 6 presents the conclusion of this study and provides information on the direction of future work.

2. Literature Review

Scheduling problems in the production process play a major role in reducing energy consumption. Production cost, production efficiency, and production quality were the main objectives of production scheduling research in the past, and green energy saving strategies were often disregarded (Duan & Wang, 2021).

Recent years have seen an increase in research on energy efficiency across a range of disciplines due to factors such as global warming, rising competition, and rising energy costs (Dauzere-Peres & Paulli, 1997). Wang et al. (2023) analyzed the fuzzy hybrid flow-shop scheduling problem on energy efficiency with an improved NSGA II algorithm. Lu et al. (2021) have developed an iterative greedy algorithm to address the energy-efficient distributed heterogeneous flow-shop scheduling problem. A reinforcement learning algorithm was developed by Zhao et al. (2023) for an energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. Wang et al. (2023) considered the energy-efficient unrelated parallel machine scheduling with general position-based deterioration. Li et al. (2016) investigated identical parallel machine scheduling problems in green manufacturing to minimize the makespan with the restriction on total energy costs. Xin et al. (2023) designed the energy-efficient unrelated parallel machine scheduling considering the general position-based deterioration. He et al. (2022) proposed solving a multi-objective energy-efficient job shop scheduling problem with a hybrid multiobjective GA (HMOGA). Flexible job shop scheduling problems were first proposed by Brucker and Schlie in 1990 and have become interesting since then (Brucker & Schlie, 1990). Brandimarte first used the concept of flexible job shop scheduling in 1993 (Brandimarte, 1993). Jiang et al. (2022) studied a wide variety of small-batch dynamic flexible job shop type scheduling on energy efficiency in the aviation business. In past research, dynamic scheduling has been proven to be effective in reducing carbon emissions, saving energy, and increasing efficiency (Wang et al., 2019). Studies on MO-FJSP in the field of energy efficiency, approaches, objectives and dynamic events are presented in Table 1.

Table 1
Studies on MO-FJSP in the field of energy efficiency, approaches, objectives and dynamic events.

Studies	Approaches	Objectives	Dynamic Events
Wenjun et al. (2016)	Pareto-based bees algorithm	Minimize the time consumption, cost, material consumption, the energy consumption, and maximize the product quality	New service request, tool replacement, failure
Nouiri et al. (2018)	PSO	Minimize the makespan and with less global energy consumption	Machine breakdowns
Li et al. (2020)	NSGA-II	Minimize the total energy consumption, makespan and employs	New job arrivals, machine breakdown
Nouiri et al. (2020)	PSO	Minimize the makespan and the energy consumption	Machine breakdowns
Duan and Wang (2020)	NSGA-II	Minimizing the total energy consumption and maximum completion time of machines	Machine breakdowns
Li et al. (2022)	Hybrid deep Q network	Minimize the makespan and total energy consumption	New job insertions and machine breakdowns
Naimi et al. (2021)	Q-learning	Minimize the makespan and energy consumption	Machine breakdowns
Duan and Wang (2022)	PSAO	Minimize the total energy consumption, the makespan and the comprehensive reusability of the system	Machine breakdowns and new job arrivals
Caldeira et al. (2020)	Backtracking search algorithm	Minimize the makespan, energy consumption, and instability	New job arrivals and turn on/off strategy
This study	**ENSGA II**	**Minimize the total energy consumption, makespan and total machine workload**	**New job arrival and rework processes.**

As a result of the literature review, we realized that there are very few dynamic event-based scheduling studies that consider energy consumption as a criterion. This study is aimed to improve the performance by adding a heuristic method to the NSGA II algorithm. This improved solution method will be called the enhanced NSGA II (ENSGA II) algorithm. In addition, the rework processes, which have not been studied as a dynamic event in this field before, will be added to the model. Thus, in manufacturing involving high raw material costs, additional operations will be added to some products that may be scrap (welding, grinding, etc.). Scrap parts will be adapted to customer requirements with rework processes, resources will be used more efficiently, and a competitive advantage will be gained in terms of sustainable production.

3. Problem description and mathematical modeling

3.1 Problem description

The traditional FJSP has been extended to account for energy consumption rework processes and new job arrivals. This problem has two stages. In the first stage, a group of jobs are scheduled on machines. In this scheduling, makespan, total machine workload and energy consumption are minimized. The second stage is the rescheduling stage. During the rescheduling stage, makespan, total machine workload and total energy consumption are minimized. At this stage, the rescheduling stage is carried out with dynamic events. The dynamic events at this stage are new job arrivals and rework processes. That is, the operations that have started before the occurrence of a dynamic event remain unaffected during the rescheduling stage. The energy consumption also varies depending on the machine’s state. There are two main items to a machine's energy items. These are processing energy (E_1) and idle energy (E_2). The sum of these two energy items presents the total energy consumption. Some assumptions are as follows for simplifying the model.

(1) Jobs are mutually independent, and pre-emption is not allowed.

(2) Every job is composed of a series of consecutive operations subject to precedence restrictions. In other words, no operation can start processing before its predecessor has finished.

(3) Only one operation can be processed at a time by each machine.

(4) At time zero, every job and every machine is available and ready, except when the new job arrives.

(5) The processing time for a machine operation is known in advance and includes setup and transportation.

(6) Every machine's energy consumption is deterministic and known in advance for every state.

(7) Except in the event of a machine breakdown, no interruptions are permitted after the process has begun.

(8) In case of part nonconformity after the job operation, the operations to be added to make the part conformity are deterministic and known in advance.

3.2 Mathematical model

The notations used to formulate the FJSSP considering rescheduling and energy consumption are first defined, followed by the MILP model.

Indices:

- a: scheduling stage, a=1,2 (1= Scheduling stage, 2= Rescheduling stage)

- i,j: jobs, i,j = 1,2,⋯, J^a

- r,s: operations, r,s = 1,2,⋯, O^a

- m: machines, m = 1,2,⋯,M

Parameters:

- J^a: number of jobs in stage a

- O^a: number of operations in stage a

- M: set of machines

- J_m^a: set of jobs that can be done on machine m at stage a ($m \in M, J_m \in J$).

- O_j^a: set of operations of job j at stage a ($j \in J, O_j \in O$).

- $S_{m,j,r}^a$: processing time of job j operation r machine m at stage a .

- $BigM$: A big number.

- $M_{j,r}$: set of machines on which operation r of job j can be performed ($j \in J, r \in O, M_{j,r} \in M$).

- $OJ_{m,j}^a$: The set of operations that can be performed on job j on machine m at stage a ($j \in J, m \in M, OJ_{m,j} \in O$).

- I_m: energy consumption per unit idle time on machine m.

- $E_{m,j,r}^a$: energy consumption per unit processing time of job j operation r on machine m at stage a.

Variables:

- z_j: completion time of job j ($\forall j \in J$).

- $\tau_{j,r}^a$: completion time of job j operation r at stage a ($\forall j \in J, \forall r \in O_j$).

- K_m^a: idle time of machine m at stage a ($\forall m \in M_{j,r}$).

- $t_{m,j,r}^a$: completion time job j operation r on machine m at stage a ($\forall j \in J, r \in O_j, \forall m \in M_{j,r}$).

- $x_{j,r,m}^a = \begin{cases} 1, \textit{ if operation r is processed on machine m at stage } a. \\ 0, \text{ otherwise} \end{cases}$ $(\forall j \in J, r \in O_j, \forall m \in M_{j,r})$.

- $y_{m,i,s,j,r}^a = \begin{cases} 1, \; if\ machine\ m\ processed\ job\ i\ operation\ s\ after\ job\ j\ operation\ r\ at\ stage\ a. \\ 0, \text{ otherwise} \end{cases}$ $\left(\forall m \in M, \forall i \in J_m, s \in OJ_{i,m}, \forall j \in J_m, \forall r \in OJ_{j,m}\right)$.

Objective Functions:

$$\min f_1 = \sum_{j \in J} z_j \tag{1}$$

$$\min f_2 = E_1 + E_2 \tag{2}$$

$$\min f_3 = \sum_{j \in J} \sum_{m \in M_{j,r}} \sum_{r \in OJ_{m,j}^a} S_{m,j,r} * x_{j,r,m}^a \tag{3}$$

Subject to:

$$E_1 = \sum_{j \in J} \sum_{m \in M_{j,r}} \sum_{r \in OJ_{m,j}^a} S_{m,j,r} * x_{j,r,m}^a * E_{m,j,r}^a \tag{4}$$

$$E_2 = \sum_{m \in M_{j,r}} K_m^a * I_m \;;\; \forall m \in M_{j,r} \tag{5}$$

$$\sum_{m \in M_{j,r}} x_{j,r,m}^a = 1;\ \forall j \in J^a, \forall r \in O_j^a \tag{6}$$

$$x_{j,r,m}^a - \sum_{i \in J_m^a} \sum_{s \in OJ_{m,i}^a} y_{m,i,s,j,r}^a \geq 0 \;;\ \forall j \in J^a, \forall r \in O_j^a, \forall m \in M_{j,r} \tag{7}$$

$$x_{i,s,m}^a - \sum_{j \in J_m^a} \sum_{r \in OJ_{m,j}^a} y_{m,i,s,j,r}^a \geq 0 \;;\ , \forall i \in J^a, \forall s \in O_j^a, \forall m \in M_{j,r} \tag{8}$$

$$\sum_{j\in J_m^a}\sum_{r\in OJ_{m,j}^a} x_{j,r,m}^a - \sum_{j\in J_m^a}\sum_{r\in OJ_{m,j}^a}\sum_{i\in J_m^a}\sum_{s\in OJ_{m,i}^a} y_{m,i,s,j,r}^a = 1\ ;\ ,\forall m \in M \tag{9}$$

$$t_{m,j,r}^a \geq S_{m,j,r} * x_{j,r,m}^a\ ;\ \forall j \in J^a, \forall r \in O_j^a, \forall m \in M_{j,r} \tag{10}$$

$$t_{m,j,r}^a - t_{m,i,s}^a + BigM * \left(1 - y_{m,i,s,j,r}^a\right) \geq S_{m,j,r}\ ;\ \forall j \in J^a, \forall r \in O_j^a, \forall i \in J^a, \forall s \in O_i^a, \forall m \in M_{j,r} \tag{11}$$

$$t_{m,j,r}^a - \tau_{j,s-1}^a \geq S_{m,j,r} - BigM * (1 - x_{m,j,r}^a)\ ;\ \forall j \in J^a, \forall r \in O_j^a, \forall i \in J^a, \forall s \in O_i^a, \forall m \in M_{j,r} \tag{12}$$

$$K_m^a \geq (t_{m,j,r}^a - t_{m,i,s}^a - S_{m,j,r}) * x_{m,j,r}^a\ ;\quad \forall j \in J^a, \forall r \in O_j^a, \forall i \in J^a, \forall s \in O_i^a, \forall m \in M_{j,r} \tag{13}$$

$$\tau_{j,r}^a \geq t_{m,j,r}^a \tag{14}$$

$$z_j \geq \tau_{j,r}^a \tag{15}$$

Objective (1) is minimizing the makespan. Objective (2) is the minimizing the total energy consumption. Objective (3) is the minimizing the total machine workload. Constraint (4) is specifically related to managing or limiting the energy consumption of a machine while it is actively processing tasks. Constraint (5) is energy consumption when a machine is idle. Constraint (6) ensures that operation of each job can be assigned to only one machine. Constraint (7) ensures that operation that may precede the operation of a job on a machine. Constraint (8) ensures that operations that may follow the operation of a job on a machine. Constraint (9) ensures that each machine has one less sequence from assigned job operations. Constraint (10) ensures that the completion time for the operation of each job is at least the same as the processing time on the relevant machine to which it is assigned. Constraint (11) ensures that the difference between the completion times of two consecutive operations is at least as much as the processing time of the following operation. Constraint (12) ensures that the difference between the completion times of one operation of each job and the previous operation is at least as much as the processing time of the following operation. Constraint (13) ensures that idle time of the machine is at least the difference between successive operations and process times on the relevant machine. Constraints (14-15) ensure that the completion time of each job is at least the completion time of the last operation of that job.

4. Proposed solution approaches

This section presents a detailed description of ENSGA-II, covering initialization, genetic processing, adaptive parameter management technique, information feedback model, encoding and decoding. To address the proposed problem, the well-known evolutionary algorithm Non-Dominant Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002) was adopted. The flow chart of the proposed ENSGA II is shown in Fig. 1, and the procedures are shown as follows.

Step 1: Initialization: Based on the updated task and machine state, a random starting population P_0 of size *S* is generated at the start of the process with iteration *gen = 0*.

Step 2: Heuristic strategy: Generate the pi value. If pi is less than D_1, the machine with the minimum processing time is selected; if pi is less than D_2, the machine with the minimum energy consumption is selected; else machine is selected as randomly ($D_1 < D_2$).

Step 3: Individual sorting: All the individuals of P_{gen} are sorted using the fast non-dominated sorting method and crowding distance sorting procedure according to the fitness values, which are the total energy consumption, total machine workload and makespan.

Step 4: Offspring generation: Generate the offspring by selection, crossover and mutation. A new population Q_{gen} is generated.

1) Selection

Selection mechanisms are employed to choose individuals from the current population to proceed to the next generation based on their fitness or objective function values. The tournament selection method is used in this article.

2) Crossover

The crossover operator's purpose is to improve the algorithm's exploration capability by appropriately generating new solutions from the current iteration.

Multi-point crossover provides a way to introduce diversity in the population of solutions and explore different regions of the search space. By exchanging multiple segments of genetic material between parent chromosomes, multi-point crossover allows for a more extensive exploration of the solution space compared to single-point crossover.

In this paper, the hybrid multi-point crossover is adopted. The rand function is used to generate random value p_c ($p_c \in [0, 1]$). If $p_c \leq 0,5$,, operation sequencing-based crossover is adopted, else machine assignment-based crossover is used as shown in Fig 3.

4. Mutation

Mutation is a genetic operator used in genetic algorithms and evolutionary algorithms to introduce genetic diversity in the population of candidate solutions. It works by making small random changes to individuals (chromosomes) in the population, allowing the algorithm to explore new regions of the search space. In this paper single-point mutations are adopted, which are randomly selected in each iteration.

Based on the provided information, here's how the mutation operation, targeting the 2nd gene in the chromosome, is performed.

An example of the mutation operation is presented in Fig. 4. The second gene was selected for mutation and this gene is $O_{8,1}$. The set of machines that can be assigned to the $O_{8,1}$ gene are M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8 and M_9. As a result of the mutation, the M_5 machine was randomly selected instead of the M_6 machine. This mutation produces a change in the processing machinery assigned to the $O_{8,1}$ task, which contributes to genetic diversity in the population. Choosing a new machine allows alternative solutions to be investigated and can potentially lead to improved performance or convergence to better solutions during the optimization process.

Step 5: Sorting new population: Combine population P_{gen} and Q_{gen} to form the new population that is R_{gen}. Following the process in step 2 and perform the fast non-dominated sorting on R_{gen}.

Step 6: Population selection: Select S individuals from R_{gen} using tournament selection to create a new population. Every member of the population is compared pairwise throughout the tournament selection process. When two individuals are compared, if one dominates, it is selected for the offspring population. Otherwise, the selected individual is the one with the larger value in the crowding distance.

Step 7: Termination: Determine if the gene of iterations reaches the maxgen. If it reaches the *maxgen*, end. Otherwise, set the *gen = gen + 1* and turn to Step 2.

Step 8: Best solution: End of iteration to select the best solution.

4.1 Encoding and decoding

In this study, a three-layer real number coding method was applied. The job and its operation sequencing, the assigned machine and the processing time on the assigned machine must be encoded in the chromosome. As shown in Fig. 5, the first layer is the process sequence, the second layer is the machine assignment, and the third layer is the processing time. The lengths of all layers are equal to the total number of operations.

Operation sequencing (OS): Genes task number explicitly encoded in it. Chromosomes are arranged from left to right, and the sequence of job numbers indicates the sequence of job processing. In the example in Figure 5, the 1st job in the 4th row represents the 2nd operation of the 1st job, that is, $O_{1,2}$.

Machine assignment (MA): MA stores an integer value equal to the machine index of the corresponding process in the compatible machine set.

Processing time (PT): Each gene is represented by an integer organized by job and operation. Each integer represents the processing time corresponding to the current operation.

The procedure of decoding is as follows;

Step 1: Obtain the chromosome generated by the operation-based encoding method.

Step 2: Search each gene from the left to the right of the chromosome to determine the operation, machine and the processing time for each gene in turn based on these chosen process plans and the job number. And obtain the index for operations r, and the index for operations on a specific machine m.

Step 3: Determine the set of job release times $R_{(t)}$, the set of machine available times $A_{(t)}$ and the set of processing time $S_{(t)}$.

Step 4: Determine the starting time of each operation on machine m.

$$If\ r = 1$$

$$F_{m,j,r} = \max\{t_{m,j,(r-1)}, R_{(t)}\}$$

$$If\ s = 1$$

$$then\ F_{m,j,r} = \max\{t_{m,j,(r-1)}, A_{(t)}\}$$

$$otherwise$$

$$F_{m,j,r} = \max\{t_{m,j,(r-1)}, t_{m,j,(s-1)}\}$$

where $F_{m,j,r}$ is the start time of job j operation r on machine m, $t_{m,j,(r-1)}$ is the completion time of job j operation (r-1) on machine m, while $t_{m,j,(s-1)}$ is the completion time of job j operation (s-1) on machine m .

Step 5: Determine the idle time of machine m.

$$If\ r > 1\ and\ s > 1$$

$$then\ T_{idle(m)} = \max\{t_{m,j,(r-1)}, t_{m,j,(s-1)}\} - t_{m,j,(s-1)}$$

$$otherwise$$

$$F_{m,j,r} = \max\{t_{m,j,(r-1)}, t_{m,j,(s-1)}\}$$

Where $T_{idle(m)}$ is the stand-by time of machine m.

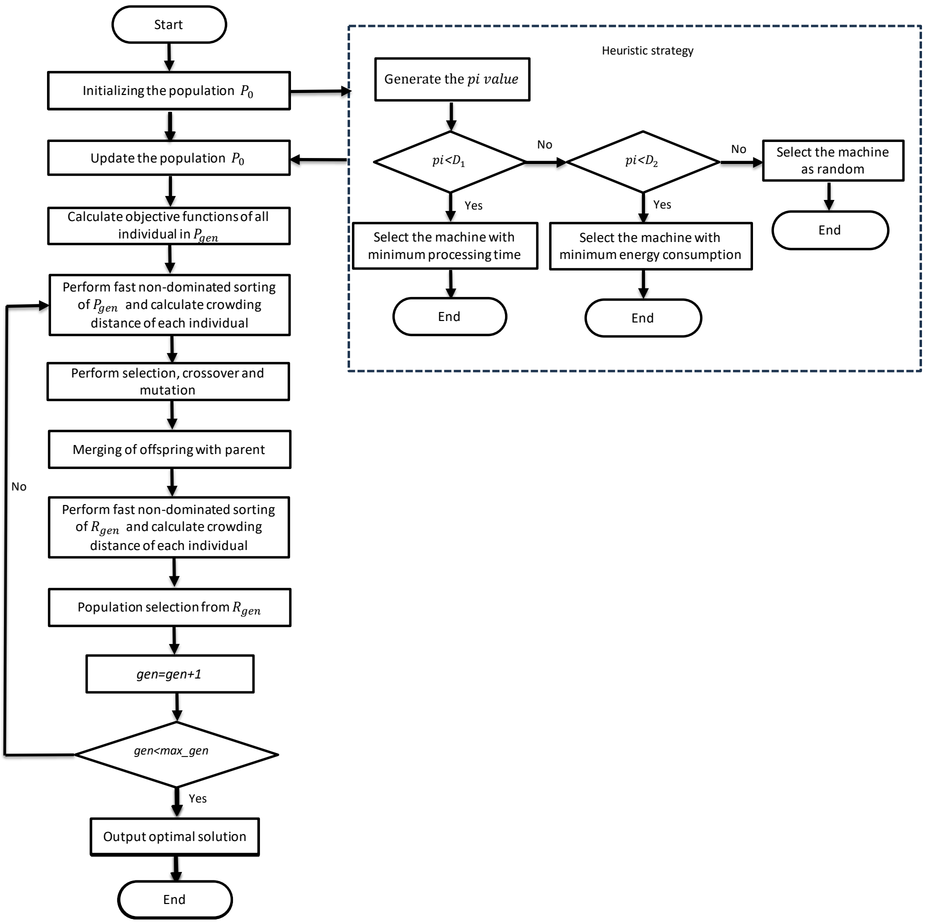

Fig. 1. Flowchart of algorithm

Step 6: Determine the completion time of every operation on machine m.

$$t_{m,j,(r-1)} = F_{m,j,r} + S_{m,j,r}$$

where $S_{m,j,r}$ is the processing time of operation job j operation r on machine m.

Step 7: Generate the sets of starting time and completion time for each operation of each job on machine m, and the $T_{idle(m)}$.

4.2 Rescheduling strategy

In this study, the rescheduling strategy proposed by Li et al. (Li et al., 2017) was used. These rescheduling strategies are shown as follows:

This strategy is rescheduling both rework processes, new jobs arriving and the existing jobs' operations that are not started at the new event insertion time. The scheduling solution for the operations of existing jobs can be changed. In this strategy, some machines may be performing some operations of existing jobs at the new job insertion time. Once machine operations are completed, the machines are available. Existing jobs are also ready to be rescheduled when the relevant processes are completed. This means that when rescheduling is applied, both machines and jobs have different start times. Therefore, this strategy must consider both machines start times and job start times for rescheduling.

5. Computational results

5.1 Parameter setting

The proposed ENSGA II was coded in Python 5.4.1 and executed on an AMD Ryzen processor at 2.3 GHz on Windows 11 operating system with 16 GB RAM.

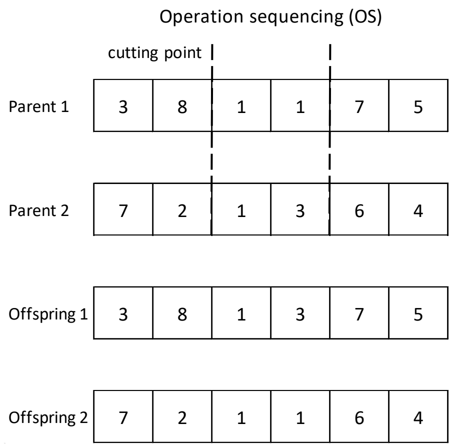

Fig. 2. Operation sequencing-based crossover

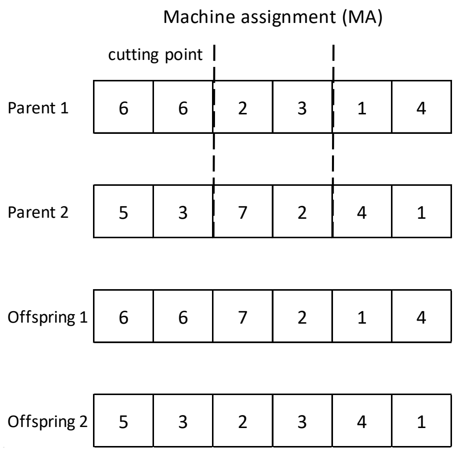

Fig. 3. Machine assignment-based crossover

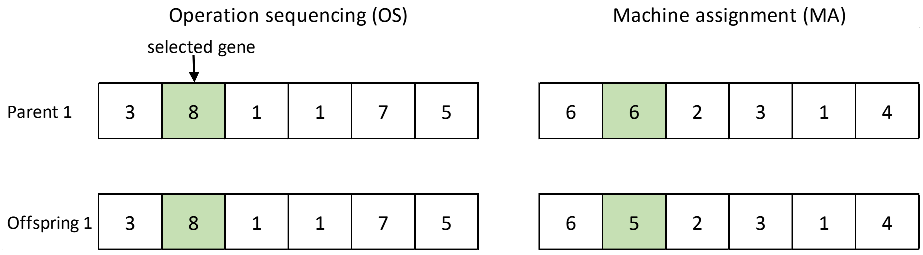

Fig. 4. Single-point mutation

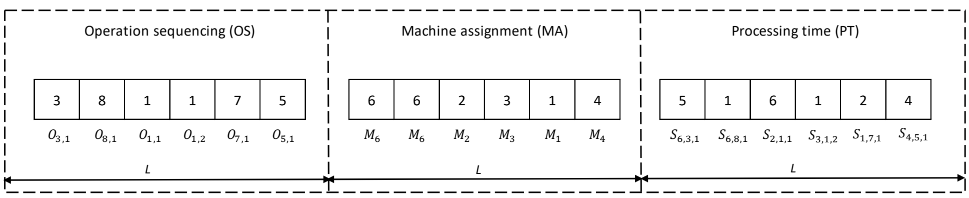

Fig. 5. Encoding diagram

The population size is assumed to be 500, the crossover rate is 0.6 and the mutation rate is 0.5, D_1 is selected as 0.33, D_2 is selected as 0.66 and the number of genes to mutate is assumed to be 1. The maximum generation is 50. Each instance is performed in 10 replicates. The Wilcoxon signed-rank test was utilized to assess the significant difference between the results performed by different metaheuristic approaches at a significance level of 0.05, to mitigate the impact of randomness on metaheuristic performance. The suggested ENSGA II performs significantly better or worse than the comparison algorithm, denoted by a "+" or "-" sign. The "=" symbol denotes the lack of a discernible difference between ENSGA II and its rivals.

5.2 Benchmark problem and data generation

In this paper, the benchmark instances are selected from the data set of Brandimarte (Brandimarte, 1993) that contains 10 instances (MK01-MK10). The values of the machine processing power are randomly generated between 0.5-1.0, idle power is randomly generated between 0.0-0.5 and shown in Table 2.

5.3 Performance metrics

To evaluate the proposed ENSGA II, we adopted the generational distance (GD) (Van & Lamont, 1998, 2000) inverted generational distance (IGD) (Zhu & Zhou, 2020) and hypervolume (HV) (Zitzler et al., 2007) as performance metrics. We normalize the non-dominated solutions that each method obtained for computational convenience.

5.3.1 GD metric

The GD metric assesses the convergence of the algorithm by calculating the average of the distance between the points in P and the nearest point in $P*$. The smaller the GD value, the better the convergence of the algorithm (Van & Lamont, 1998). The GD metric is formulated as follows,

$$GD(P,P^{*}) = \frac{\sqrt{\sum_{y \in P} min_{y \in P^{*}} dis(x,y)^2}}{|P|} \tag{16}$$

where the dis(x, y) represents the Euclidean distance between point y in solution set P and point × in the reference set P*.

5.3.2 IGD metric

The IGD metric assesses the convergence and diversity of the algorithm by calculating the average of the distance between the points in $P*$ and the nearest point in P (Zhu & Zhou, 2020). The IGD metric is formulated as;

$$IGD(P,P^{*}) = \frac{\sqrt{\sum_{x \in P^{*}} min_{y \in P^{*}} dis(x,y)^2}}{|P^{*}|} \tag{17}$$

IGD measures how close the solutions in the pareto front approximation are to the true Pareto front. It provides a quantitative assessment of the quality of the approximation, allowing researchers and practitioners to compare different algorithms and parameter settings in multi-objective optimization. Lower values of IGD indicate better approximation quality, meaning that the Pareto front approximation is closer to the true pareto front.

Table 2
Energy consumption on processing mode and idle mode for each machine

Machine No.	Energy consumption on processing mode	Energy consumption on idle mode
Machine 1	0.85	0.18
Machine 2	0.72	0.17
Machine 3	0.97	0.12
Machine 4	0.76	0.36
Machine 5	0.97	0.21
Machine 6	0.80	0.39
Machine 7	0.60	0.50
Machine 8	0.62	0.48
Machine 9	0.98	0.47
Machine 10	0.85	0.32
Machine 11	0.96	0.32
Machine 12	0.69	0.39
Machine 13	0.55	0.48
Machine 14	0.95	0.03
Machine 15	0.57	0.19

5.3.3 HV metric

The hypervolume indicator is widely used in evolutionary multi-objective optimization algorithms to guide the search process. By maximizing the hypervolume, these algorithms aim to find a diverse set of high-quality solutions that cover as much of the objective space as possible (Zitzler et al., 2007). The HV metric is formulated as the following,

$$HV = \frac{\sum_{s\in S} V_S}{|\mathrm{S}| \; x \; V_T} \tag{18}$$

$\sum_{s\in S} V_S$, where V_S is the hypercube of Sin order concerning the reference point. Since the hypervolume can lead to large values which corresponds to the ratio of the total volume V_T covered by the reference point and the origin point.

5.4 Performance analysis of ENSGA II

The proposed ENSGA II performance is compared with three multi-objective optimization methods: AGEMOEA2 (Annibale, 2019), SMS-EMOA (Nicola et al., 2007), SPEA2 (Zitzer et al., 2001) based on GD, IGD and HV performance metrics. All the compared algorithms had their population size and maximum number of iterations set to 1000 and 50, respectively, to enable a fair comparison of the findings. Each instance was performed thirty times independently due to the inherent unpredictability involved. Table 3's objective function comparison of the algorithms reveals that ENSGA II outperforms the other three algorithms in terms of getting a greater number of non-dominated solutions. Fig. 6 shows the Gantt chart of the MK04 instance obtained by ENSGA II, and Fig. 7 shows the pareto fronts obtained by AGEMOEA2, SMS-EMOA, SPEA2, and ENSGA II. As shown in Table 4, ENSGA II obtains better results than AGEMOEA2, SMS-EMOA and SPEA2. ENSGA II obtains a smaller GD than AGEMOEA2, SMS-EMOA and SPEA2 in all instances. This shows that the ENSGA II algorithm is superior to the other three algorithms in all instances. Fig. 8 is the box plot based on the GD results in Table 4. Regarding the IGD metric results presented in Table 5, the ENSGA II algorithm obtained better than the other 3 algorithms. The ENSGA II algorithm is superior to the other 3 algorithms except for 4 test instances. Table 6 shows the HV-based results obtained by ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2. In Table 6, the HV of ENSGA II is larger than the others except for four instances. Fig. 10 is the boxplot based on the ENSGA II results in Table 6. The results show that the ENSGA II is more suitable for minimizing the makespan, energy consumption and total machine workload. ENSGA II has better performance than AGEMOEA2, SMS-EMOA and SPEA2 in solving the MO-FJSP.

Table 3
Comparison between the algorithms in terms of objective functions

	Makespan				Energy Consumptions				Machine Load			
Benchmark instances	*E-NSGA II*	*AGEMOEA2*	*SMS-EMOA*	*SPEA2*	*E-NSGA II*	*AGEMOEA2*	*SMS-EMOA*	*SPEA2*	*E-NSGA II*	*AGEMOEA2*	*SMS-EMOA*	*SPEA2*
MK01	**46**	57	51	56	**136.8**	178.8	168.4	166.2	**153**	174	173	180
MK02	**25**	40	39	47	**88**	150.8	151.9	153.1	**106**	160	165	156
MK03	258	266	**252**	266	**1049.8**	1051.3	1075.8	1077.8	1075	1087	1111	**1064**
MK04	**101**	105	104	109	399.4	397.3	**381.9**	413.2	**367**	374	397	391
MK05	**200**	222	211	**200**	**587.9**	599.2	590.9	591.1	697	691	**689**	695
MK06	**113**	122	117	132	553.5	570.5	**553**	621.1	**476**	498	518	490
MK07	200	209	**199**	203	**670.1**	712.2	731.5	679.8	**765**	790	843	773
MK08	**565**	585	576	585	2716.8	2639.5	**2628.9**	2786.5	2602	2623	**2578**	2639
MK09	**437**	455	480	456	**2283.1**	2447	2608.4	2560.8	**2377**	2459	2433	2402
MK10	382	393	**361**	374	**2264.6**	2388	2344.6	2338.3	2171	2205	**2135**	2164
+/-/=	**6/3/1**	0/10/0	3/7/0	0/9/1	**7/3/0**	0/10/0	3/7/0	0/10/0	**6/4/0**	0/10/0	3/7/0	1/9/0

Table 4
Mean and standard deviation (std) value of GD metric obtained by ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Instances	Size	*E-NSGA II*		*AGEMOEA2*		*SPEA2*		*SMS-EMOA*	
	m × n	*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
MK01	6 × 10	**0.137**	0.117	0.325	0.244	0.315	0.241	0.340	0.246
MK02	6 × 10	**0.394**	0.338	0.689	0.304	0.607	0.246	0.546	0.385
MK03	8 × 15	**0.258**	0.153	0.282	0.096	0.332	0.232	0.377	0.333
MK04	8 × 15	**0.304**	0.206	0.376	0.477	0.359	0.128	0.357	0.231
MK05	4 × 15	**0.057**	0.064	0.352	0.312	0.382	0.337	0.076	0.036
MK06	15 × 10	**0.175**	0.154	0.279	0.234	0.249	0.271	0.189	0.142
MK07	5 × 20	**0.125**	0.101	0.405	0.361	0.177	0.195	0.212	0.123
MK08	10 × 20	**0.087**	0.053	0.153	0.198	0.099	0.087	0.162	0.232
MK09	10 × 20	**0.105**	0.066	0.106	0.104	0.439	0.412	0.116	0.066
MK10	15 × 20	**0.059**	0.060	0.277	0.149	0.132	0.042	0.190	0.311
+/-/=		10/0/0		0/10/0		0/10/0		0/10/0	

5.5 Experiment on Rescheduling

In this section MK01 and MK02 are employed as the test instance. Table 7 shows the newly inserted job to MK01 instance. Table 8 shows the rework processes to MK02 instances. A rework process is a new operation or set of operations added between 2 operations. Rework processes consisting of 2 operations have been added between the 5th operation and the 6th operation in the 4th job of the MK02 example in Table 8. Thus, job number 4 of the MK04 example, which consists of 6 operations, has become a job consisting of 8 operations instead of 6 operations.

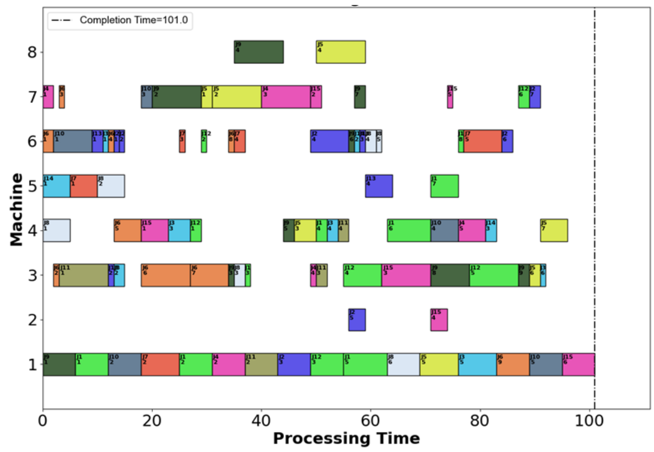

Fig. 6. Gantt chart of MK04 instance

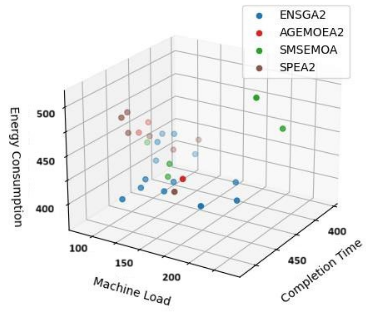

Fig. 7. Pareto fronts obtained by ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Fig. 13 is the scheduling scheme in the case of no rework processes. Fig. 14 represents the new scheduling scheme obtained by adding the rework processes to Job 4 when the time is "12" and rescheduling the rework processes with the unfinished operation in the original scheduling scheme. To avoid scrapping the part in Job 4 at the 15th unit time, 2-stage rework processes were added between the 5th and 6th operations.

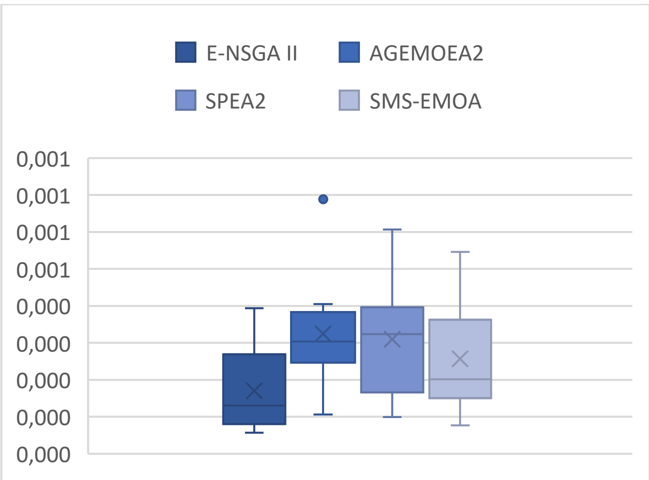

Fig. 8. Boxplots of GD values obtained by the ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

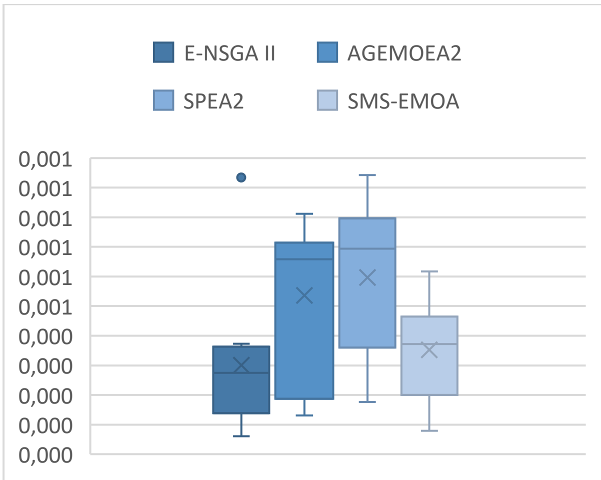

Fig. 9. Boxplots of IGD values obtained by the ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Table 5
Mean and standard deviation (std) value of IGD metric obtained by ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Instances	Size	*E-NSGA II*		*AGEMOEA2*		*SPEA2*		*SMS-EMOA*	
	m × n	*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
MK01	6 × 10	**0.261**	0.176	0.672	0.137	0.716	0.151	0.618	0.209
MK02	6 × 10	**0.153**	0.272	0.605	0.133	0.436	0.041	0.413	0.127
MK03	8 × 15	0.359	0.076	**0.131**	0.157	0.203	0.153	0.551	0.112
MK04	8 × 15	0.373	0.029	**0.141**	0.021	0.412	0.276	0.214	0.009
MK05	4 × 15	0.934	0.058	0.700	0.108	0.742	0.116	**0.080**	0.055
MK06	15 × 10	**0.289**	0.179	0.694	0.186	0.671	0.155	0.378	0.154
MK07	5 × 20	**0.060**	0.044	0.758	0.068	0.902	0.052	0.160	0.072
MK08	10 × 20	**0.168**	0.117	0.812	0.135	0.176	0.155	0.437	0.103
MK09	10 × 20	0.310	0.037	**0.203**	0.087	0.943	0.059	0.310	0.108
MK10	15 × 20	**0.094**	0.051	0.644	0.126	0.761	0.114	0.367	0.042
+/-/=		6/4/0		3/7/0		0/10/0		1/9/0	

Table 6
Mean and standard deviation (std) value of HV metric obtained by ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Instances	Size	*E-NSGA II*		*AGEMOEA2*		*SPEA2*		*SMS-EMOA*	
	m × n	*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
MK01	6 × 10	0.089	0.094	0.284	0.343	**0.235**	0.250	0.035	0.052
MK02	6 × 10	**0.833**	0.372	0.809	0.302	0.648	0.387	0.667	0.471
MK03	8 × 15	**0.341**	0.110	0.034	0.046	0.118	0.122	0.245	0.370
MK04	8 × 15	0.184	**0.263**	0.176	0.181	0.000	0.000	0.122	0.149
MK05	4 × 15	0.025	0.045	0.167	0.200	**0.277**	0.328	0.008	0.011
MK06	15 × 10	0.068	0.101	0.116	0.112	**0.232**	0.371	0.022	0.028
MK07	5 × 20	**0.206**	0.113	0.163	0.213	0.000	0.000	0.035	0.062
MK08	10 × 20	**0.479**	0.209	0.124	0.157	0.085	0.110	0.312	0.275
MK09	10 × 20	**0.173**	0.070	0.099	0.105	0.047	0.094	0.035	0.046
MK10	15 × 20	**0.432**	0.103	0.381	0.352	0.000	0.000	0.171	0.184
+/-/=		6/4/0		1/9/0		3/7/0		0/10/0	

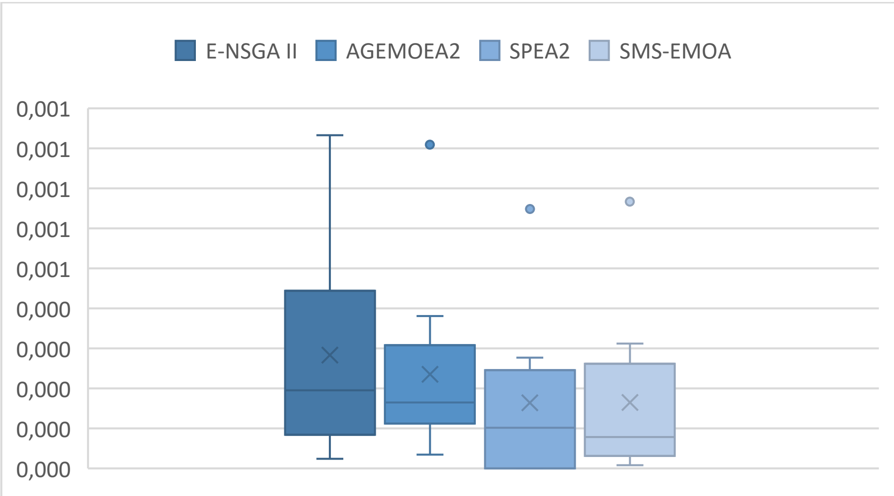

Fig. 10. Boxplots of HV values obtained by the ENSGA II, AGEMOEA2, SMS-EMOA and SPEA2

Thus, the number of operations for Job 4 increased from 6 to 8. With the new rework processes, the makespan did not change, the total machine workload increased by 24 units and the total energy consumption increased by 19.9 units. Fig. 12 and Fig. 14 show that the ENSGA II algorithm can deal with the multi-objective flexible job shop scheduling problem considering new job inserting and rework processes, proving the ENSGA II algorithm's practicability.

Table 7
Information about the newly inserted job to MK01 instance

New Job	**Inserting Time**	**Operation**	**M_1**	**M_2**	**M_3**	**M_4**	**M_5**	**M_6**
Job 11	30	$O_{11,1}$	4	-	3	6	-	-
		$O_{11,2}$	-	1	-	-	2	5
		$O_{11,3}$	5	6	3	-	2	1
		$O_{11,4}$	-	1	2	1	-	-
		$O_{11,5}$	-	3	-	3	5	4
		$O_{11,6}$	-	5	2	4	-	3
		$O_{11,7}$	2	-	1	5	3	-
		$O_{11,8}$	1	-	3	3	-	-

Table 8
Information about the rework processes to MK02 instance

Rework Job	**Inserting Time**	**Operation (After rework event)**	**Operation (Before rework event)**	**M_1**	**M_2**	**M_3**	**M_4**	**M_5**	**M_6**
Job 4	15	$O_{4,1}$	$O_{4,1}$	4	3	5	-	2	3
		$O_{4,2}$	$O_{4,2}$	3	4	-	2	6	1
		$O_{4,3}$	$O_{4,3}$	-	6	-	-	-	-
		$O_{4,4}$	$O_{4,4}$	1	6	3	3	6	5
		$O_{4,5}$	$O_{4,5}$	4	3	-	5	4	3
		$O_{4,6}$	-	-	2	3	-	1	4
		$O_{4,7}$	-	4	-	2	1	3	-
		$O_{4,8}$	$O_{4,6}$	5	4	3	1	5	3

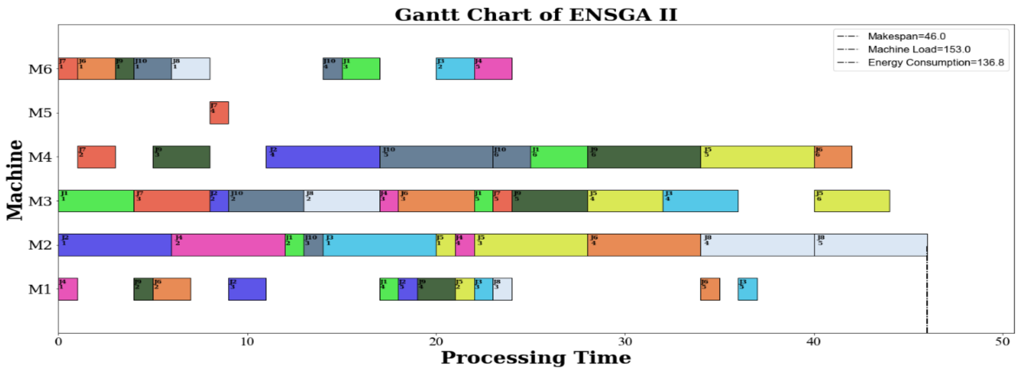

Fig. 11. The original scheduling scheme of the MK01 obtained by ENSGA II

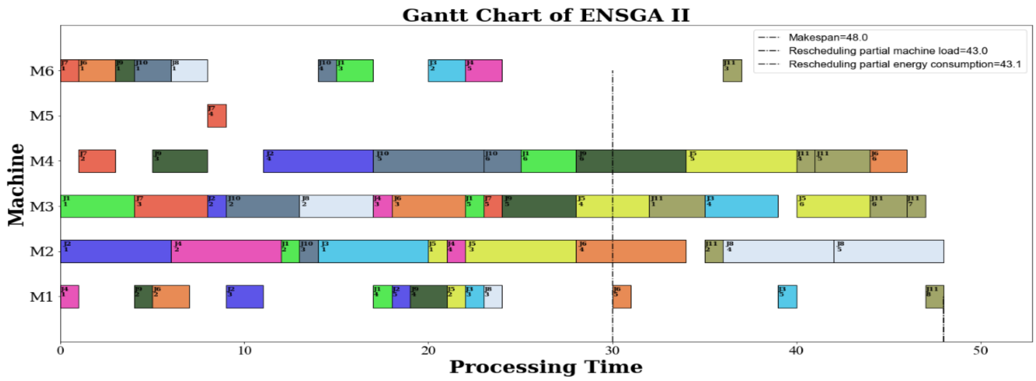

Fig. 12. The rescheduling scheme of the MK01 obtained by ENSGA II

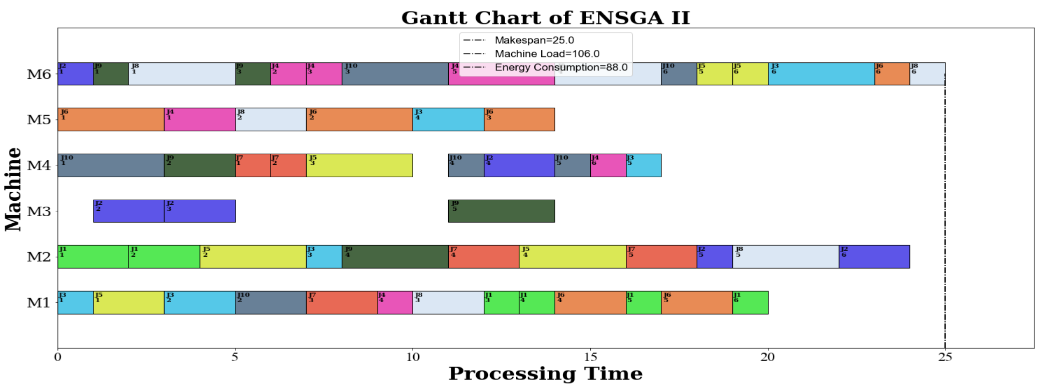

Fig. 13. The original scheduling scheme of the MK02 obtained by ENSGA II

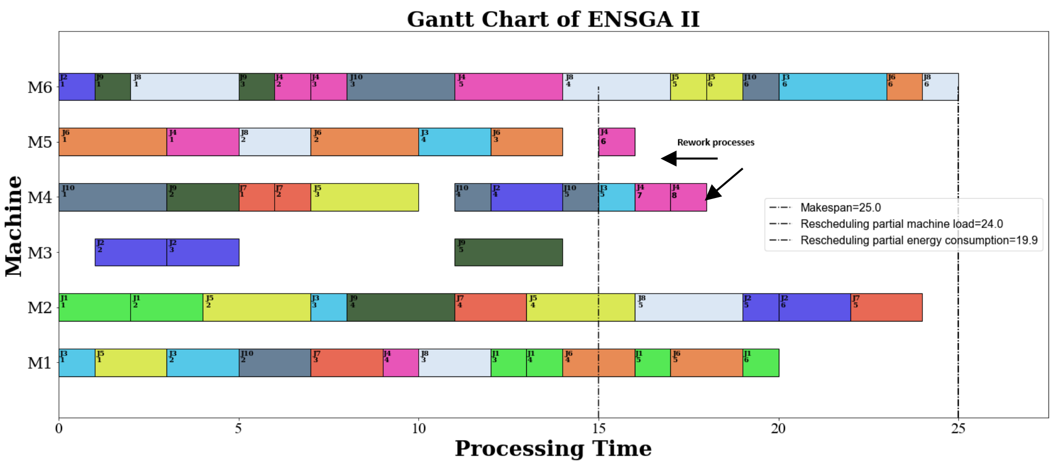

Fig. 14. The rescheduling scheme of the MK02 obtained by ENSGA II

6. Conclusions

This paper proposes enhanced NSGA II algorithm to apply energy efficiency multi-objective flexible job shop scheduling problems considering new job arrival and rework processes. The objective is to minimize the makespan, total machine workload and total energy consumption. The performance of the proposed algorithm was calculated and evaluated using GD, IGD and HV performance metrics on the test instances. Experimental results show that the performance of the proposed algorithm outperforms some well-known multi-objective algorithms: AGEMOEA2, SMS-EMOA and SPEA2. According to these results, the proposed algorithm can greatly help companies manage their business plans in terms of sustainable production.

In the future, we will focus on the following subjects: (1) Improving the performance of the algorithm apply to machine learning techniques (2) To study the energy efficiency scheduling for the flexible job shop problem under multiple resource constraints such as worker performance, machine setup time, and transportation time. (3) The proposed algorithm can be applied to real-world scheduling problems.

References

Annibale, P. (2019). An adaptive evolutionary algorithm based on non-Euclidean geometry for many-objective optimization. *In Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, 595–603.

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by taboo search. *Annals of Operations Research*, *41*,157–183.

Brucker, P., & Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing, 45*, 369–375.

Caldeira, R.H., Gnanavelbabu, A., & Vaidyanathan, T. (2020). An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. *Computers & Industrial Engineering*, 149, 106863.

Dauzere-Peres, S., & Paulli, J. (1997). An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research*, *70,* 281-306.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

Duan, J.G., & Wang, J.H. (2021). Energy-efficient scheduling for a flexible job shop with machine breakdowns considering machine idle time arrangement and machine speed level selection. *Computers & Industrial Engineering, 161*, 1-14.

Duan, J., & Wang, J. (2022). Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence. *Expert Systems with Applications, 203*, 117489.

He, L., Chiong, R., Li W., Dhakal, S., Cao, Y., & Zhang, Y. (2022). Multiobjective Optimization of Energy-Efficient JOB-Shop Scheduling With Dynamic Reference Point-Based Fuzzy Relative Entropy. *IEEE Transactions on Industrial Informatics*, *18*(1), 600-610

Jiang, X.Y., Tian, Z.Q., Liu, W., & Li, Z. (2022). Energy-efficient scheduling of flexible job shops with complex processes: A case study for the aerospace industry complex components in China. *Journal of Industrial Information Integration*, *27*, Article 100293.

Li, K., Zhang, X., Leung, J. Y. T., & Yang, S.L. (2016). Parallel Machine Scheduling Problems in Green Manufacturing Industry. *Journal of Manufacturing Systems, 38*, 98–106.

Li, X. Y., & Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *International Journal of Production Economics*, *174*, 93–110.

Li, X., Peng, Z., Du, B., Guo, J., Xu, W., & Zhuang, K. (2017). Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems. *Computers & Industrial Engineering, 113*, 10–26.

Li, Y., He, Y., Wang, Y., Tao, F., & Sutherland J.W. (2020). An optimization method for energy-conscious production in flexible machining job shops with dynamic job arrivals and machine breakdowns. *Journal of Cleaner Production, 254*, 120009.

Li, Y., Gu, W., Yuan, M., & Tang, Y. (2022). Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network. *Robotics and Computer-Integrated Manufacturing*, *74*, 102283.

Lu, C., Gao, L., Yi, J., & Li, X. (2020). Energy-efficient scheduling of distributed flow shop with heterogeneous factories: A real-world case from automobile industry in China. *IEEE Transactions on Industrial Informatics*, *17*(10), 6687-6696.

Naimi, R., Nouiri, M., & Cardin, O. (2021). A Q-learning rescheduling approach to the flexible job shop problem combining energy and productivity objectives. *Sustainability*, *13*(23), 13016.

Nicola, B., Boris, N., & Michael, E. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669.

Nouiri, M., Bekrar, A., & Trentesaux, D. (2018). Towards energy efficient scheduling and rescheduling for dynamic flexible job shop problem. *In: 16th IFAC symposium on information control problems in manufacturing* INCOM 2018, vol. 51, no. 11. p. 1275–80.

Nouiri, M., Bekrar, A., & Trentesaux, D. (2020). An energy-efficient scheduling and rescheduling method for production and logistics systems. *International Journal of Production Research*, *58*(11), 3263-3283.

Xin, X., Jiang, Q., Li, C, Li, S., & Chen, K. (2023). Permutation flow shop energy-efficient scheduling with a position-based learning effect. *International Journal of Production Research*, *61*(2), 382-409

Xu, W., Shao, L., Yao, B., Zhou, Z., & Pham, D. T. (2016). Perception data-driven optimization of manufacturing equipment service scheduling in sustainable manufacturing. *Journal of Manufacturing Systems*, *41*, 86-101.

Van, V. D. A., & Lamont, G. B., (2000). On measuring multi-objective evolutionary algorithm performance. *in Proc. 2000 Congress on Evolutionary Computation*, La Jolla, USA, 2000, 204–211.doi:10.1109/CEC.2000.870296.

Van, V. D. A., & Lamont, G. B. (1998). Evolutionary computation and convergence to a Pareto front. *in Proc. Late-breaking Paper at the Genetic Programming 1998 Conference*.

Wang, K., Wu, M., Sun, Y., Shi, X., Sun, A., & Zhang, P. (2019). Resource abundance, industrial structure, and regional carbon emissions efficiency in China. *Resources Policy*, *60*, 203-214.

Wang, Y., Che, A., & Feng, J. (2023). Energy-efficient unrelated parallel machine scheduling with general position-based deterioration, *International Journal of Production Research*, *61*(17), 5886-5900.

Wang, Y.J., Wang, G.-G., Tian, F.-M., Gong, D.-W., & Pedrycz, W. (2023). Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. *Engineering Applications in Artificial Intelligence, 121*, Article 105977.

Zhao, F., Jiang, T., & Wang, L. (2022). A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. *IEEE Transactions on Industrial Informatics*, *19*(7), 8427-8440.

Zitzer, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *In: Proceedings of the EROGEN Conference*, pp. 182–197.

Zitzler, E., Brockhoff, D., & Thiele, L. (2007). The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration. *Evolutionary multi-criterion optimization*. Springer, pp. 862–876

Zhu, Z., & Zhou, X. (2020). An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Computers and Industrial Engineering*, 140(January), Article 106280.