Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8035806, 12 pages
https://doi.org/10.1155/2021/8035806

Research Article

Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming

Andy Ham,[1] **Myoung-Ju Park**,[2] **and Kyung Min Kim**[3]

[1] *Applied Engineering Technology, North Carolina A&T State University, Greensboro, NC 27411, USA*
[2] *Industrial and Management Systems Engineering, Kyung Hee University, Seoul, Republic of Korea*
[3] *Industrial Management and Engineering, Myong Ji University, Seoul, Republic of Korea*

Correspondence should be addressed to Myoung-Ju Park; pmj0684@khu.ac.kr

Received 7 April 2021; Revised 16 June 2021; Accepted 22 June 2021; Published 30 June 2021

Academic Editor: Snežana Tadić

Compromising productivity in exchange for energy saving does not appeal to highly capitalized manufacturing industries. However, we might be able to maintain the same productivity while significantly reducing energy consumption. This paper addresses a flexible job shop scheduling problem with a shutdown (on/off) strategy aiming to minimize makespan and total energy consumption. First, an alternative mixed integer linear programming model is proposed. Second, a novel constraint programming is proposed. Third, practical operational scenarios are compared. Finally, we provide benchmarking instances, CPLEX codes, and genetic algorithm codes, in order to promote related research, thus expediting the adoption of energy-efficient scheduling in manufacturing facilities. The computational study demonstrates that (1) the proposed models significantly outperform other benchmark models and (2) we can maintain maximum productivity while significantly reducing energy consumption by 14.85% (w/o shutdown) and 15.23% (w/shutdown) on average.

1. Introduction

Energy consumption is a very important issue for our society in terms of both environment and economy. U.S. Energy Information Administration [1] recently reported international energy outlook 2019 with projections to 2050. In the report, the world energy consumption is expected to increase by approximately 50%, accounting for more than half of the non-OECD Asian countries, including China and India. Industrial sectors, such as manufacturing, agriculture, and construction, are the largest consumer among end users, and their energy consumption will increase by more than 30% from 2018 to 2050. The share of energy consumed by energy-intensive manufacturing holds steady at 50% from 2018 to 2050. These predictions and phenomena were analyzed in several studies [2–4].

Manufacturing has enormous potential for energy saving because 80% of the energy consumed by machines occurs in the idle state [5]. For this reason, several methods to increase energy efficiency in manufacturing have been attempted. The common method is to optimize a production schedule by considering green metrics along with traditional performance indicators. This approach called energy-aware scheduling or energy-efficient scheduling has the advantage of achieving significant performance at no extra cost [6].

In this paper, we deal with energy-efficient scheduling in the flexible job shop scheduling problem (FJSP). Mouzon et al. [5] observed that if an idle period is long enough, the energy could be saved by using shutdown (on/off) strategy to turn the machine off and on. Che et al. [7] first developed a mixed integer linear programming (MILP) model for a single machine scheduling with a shutdown strategy, and the model was effectively validated with CPLEX solver. Subsequently, the shutdown strategy was applied to various production systems, and the first MILP model for FJSP with a shutdown strategy was proposed by Zhang et al. [8]. However, the proposed model was complex and computationally inefficient, so Meng et al. [9] improved it. However, both studies used only one type of MILP formulations for FJSP. This paper proposes the most efficient MILP formulation.

The contributions of this paper are threefold. First, we propose an alternative MILP model for FJSP with a shutdown strategy. Second, we devise a constraint programming (CP). Third, the experimental study demonstrates that we can reduce energy consumption by 14.85% (w/o shutdown) and 15.23% (w/shutdown) on average, while not compromising productivity. In addition, we provide benchmarking instances, CPLEX MIP and CP source codes, and GA code, in order to promote related-research, thus expediting the adoption of energy-efficient scheduling in manufacturing facilities.

The remainder of the paper is organized as follows: Section 2 shows the literature review of FJSP, energy-efficient scheduling, and constraint programming. In Section 3, the problem description and solution methods are described. In Section 4, we compare the performance of proposed models and existing model through computational experiments. Finally, the conclusion and some directions for future research are given in Section 5.

2. Literature Review

2.1. Flexible Job Shop Scheduling Problem. The FJSP has been extensively studied over the past 30 years. A variety of techniques from exact methods to heuristics have been used in this research. Exact methods include branch-and-bound algorithm, MILP, and Lagrangian relaxation method among others, while heuristics include ant colony optimization, artificial bee colony, artificial immune system, evolutionary algorithms, greedy randomized adaptive search procedure, neighborhood search, particle swarm optimization, simulated annealing, Tabu search, hybrid techniques, and so on. See Chaudhry and Khan [10] for the survey about FJSP. We focus on the MILP method.

MILP models for FJSP are classified into four different types by the main binary decision variable. The first is the machine-position formulation (MPF) that determines the position of the machine, where each operation is processed [8, 9, 11, 12]. The second is the general precedence formulation (GPF) that determines whether one operation precedes the other operation on the same machine [13, 14]. Note that one operation is not necessarily positioned immediately before the other operation. The third is the immediate precedence formulation (IPF) that determines whether one operation immediately precedes the other operation on the same machine [15]. The fourth is the time indexed formulation (TIF) that determines the time of the machine when each operation is started [16]. Demir and Kürşat Isleyen [17] classified mathematical models by this criterion and compared the computational results of MILP models.

2.2. Energy-Efficient Scheduling. The studies for energy-efficient scheduling are classified into four groups based on saving methods. The first is to reduce unnecessary idle time. This is a model that extends the existing model to save energy by reducing unnecessary idle time. The second is to shut down the idle machines. Turning off machines during the idle can save energy when idle time is long enough. The decision when to shut down a machine is added into the traditional scheduling model. The third is to slow down a machine speed. Energy consumption depends on a speed of a machine. We can save energy by adjusting a speed of a machine without impacting the makespan. The fourth is off-peak production. In the peak time, electricity costs are high. Therefore, a production at nonpeak time can save a significant amount of electricity costs. Table 1 categorizes studies on energy-efficient scheduling by this criterion. See Gahm et al. [37] and Gao et al. [38] for reviews of energy-efficient scheduling.

Our study belongs to the energy-efficient FJSP using the shutdown option. Among the studies, Dai et al. [25] and Wu and Sun [26] developed a genetic simulated annealing algorithm. Zhang et al. [8] developed MILP model and discovered energy-efficient rules that could be implemented in real practice. Meng et al. [9, 27] proposed several effective MILP models and evaluated the performance of these models. Zhang et al. [8] and Meng et al. [9, 27] used MPF, while our paper uses IPF. Moreover, we develop constraint programming and genetic algorithm models.

2.3. Constraint Programming. Hiller and Lieberman [39] noted that no presentation of the basic ideas of MILP is complete these days without introducing CP––that is promising to greatly expand our ability to formulate and solve various scheduling problems. CP has been applied to various scheduling problems, demonstrating a rapid computational speed. The search within IBM CPLEX CP Optimizer is equipped with the presolve functionality, some constraint propagation algorithms, temporal linear relaxation used to guide the search, and two search space exploration strategies that are used concurrently: the large neighborhood search for producing good quality solutions and failure-directed search for proving infeasibility or optimality [40].

3. Problem Description and Solutions

Consider a flexible job shop environment that consists of a set of heterogeneous machines ($k \in K$) and a set of jobs ($i \in I$). Each job i consists of a set of operations ($J_i = \{1, \ldots, n_i\}$). Each operation needs to be processed in a specific order (known as precedence constraints) for a given job. A machine can perform at most one operation at a time. Each operation must be processed by one of qualified machines. In addition to the standard FJSP problem, an amount of energy consumption is considered in this energy-aware scheduling approach. The energy is consumed during production, idle, and shutdown. The production requires the highest amount of energy, while the idle does the medium and the shutdown requires the lowest. In particular, the shutdown can be enforced when a continuous idle period of a machine is expected to be long enough to compensate the shutdown penalty. When idle times are inevitable, the key is to adjust small-size idle intervals and locate them in a single large-size interval to turn on a long-size shutdown.

Table 1: Articles including energy-efficient scheduling models.

Saving method	Environment	References
Reduce idle time	Single machine	Jiang et al. [18]
	Hybrid flow shop	Li et al. [19]; Zhang et al. [20]
	JSP	Jiang et al. [21]
	FJSP	Yin et al. [22]; Jiang et al. [23]
Shutdown idle machines	Single machine	Mouzon et al. [5]; Che et al. [7]
	Hybrid flow shop	Meng et al. [24]
	Meta Heuristic	Dai et al. [25]; Wu and Sun [26]
	FJSP MILP-MPF	Zhang et al. [8]; Meng et al. [9]; Meng et al. [27]
	MILP-IPF	This study
Exploit variable machine speed	Single machine	Che et al. [28]
	Flow shop	Fang et al. [29]; Mansouri et al. [30]
	JSP	Zhang and Chiong [31]
	FJSP	Zhang et al. [32]
Shift production soft-enter to off-peak	Single/parallel machine	Rager et al. [33]
	Hybrid flow shop	Schulz et al. [34]
	FJSP	Moon and Park [35]; Gong et al. [36]

3.1. Mixed Integer Linear Programing Models. Meng et al. [9] presented six MILP models for our problem and showed that the second model outperforms all the other models through the numerical experiments. We term this efficient model MILP-2. Despite the superiority of MILP-2, we found that it can be further improved during the implementation by employing tuples instead of arrays, thus dramatically reducing the number of binary decision variables. For instance, the MILP-2 uses a binary variable $Y_{i,j,k,t}$ that represents whether the j-th operation of job i is processed at the t-th position of machine k. This variable is a sparse array, in which most of the elements are zero. Moreover, Meng et al. [9] set the maximum number of the positions of machine k (denoted by m_k) to the total number of operations ($\sum_{i\in I} n_i$), which is excessively large. We set m_k to the number of operations that machine k can process in the instance. By taking those measures, we were able to reduce the number of binary variables by 50–60% in the same instances. We term this efficient model MILP-2A.

MILP-2 is an extension of MPF for FJSP so that energy consumption can be considered. However, Demir and Kürşat Isleyen [17] showed that MPF is the slowest model among alternative models. Choi and Choi [15] proposed a new immediate precedence formulation (IPF) that decides whether one operation immediately precedes the other as shown in Figure 1.

Now, we will propose a new model that improves IPF and extends it to account for energy consumption. We mostly use the same notations as Meng et al. [9]. The details are as follows.

Parameters:

I: set of jobs.

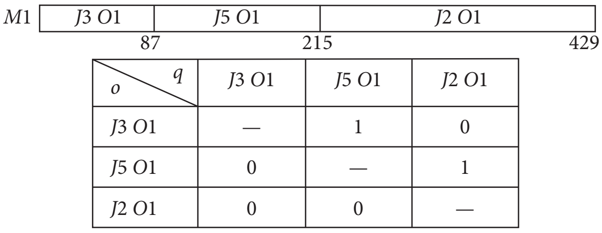
n_i: number of operations of job $i \in I$.

J_i: set of operations of job $i \in I$, that is, $J_i = \{1, \ldots, n_i\}$.

O: set of job and operation pairs, that is, $O = \{(i, j) | i \in I,\ j \in J_i\}$.

K: set of all machines.

K_o: set of machines which can process operation $o \in O$.



o \ q	J3 O1	J5 O1	J2 O1
J3 O1	—	1	0
J5 O1	0	—	1
J2 O1	0	0	—

Figure 1: An illustration of immediate precedence formulation using a decision variable $W_{o,q}$.

$p_{o,k}$: processing time of $o \in O$ by machine $k \in K_o$.

D_k: the unit energy consumed at the idle time for machine k when the shutdown strategy is not used

P_0: the unit energy consumed by a facility for lighting, heating, and cooling during the makespan.

$E_{o,k}$: the energy consumed when operation o is processed by machine k.

N_k: the maximum times of shutdown strategy for machine k.

G_k: the energy consumption of machine k when the shutdown strategy is used.

TB_k: the breakeven period of machine k in which the same amount of energy is consumed during the idle time whether the shutdown strategy is used or not. Hence, $TB_k = G_k / D_k$.

O_q: the set of operations which can be processed by the same machine with operation $q \in O$.

PO: the set of precedence operation pairs in the same job operations, that is, $PO = \{(o, q) | o = (i,\ j), q = (i, j + 1),\ i \in I,\ j = 1, \ldots, n_i - 1\}$.

$M_1 = \sum_{o\in O} \max\{p_{o,k} | k \in K_o\}$.

$M_2 = \max\{D_k \cdot TB_k | k \in K\}$.

Decision variables:

$X_{o,k}$: 1 if $o \in O$ is processed by machine $k \in K_o$ and 0 otherwise.

S_o: start time of $o \in O$.

C_o: completion time of $o \in O$.

$C_{\max}$: makespan, that is, $C_{\max} = \max\{c_o | o = (i, n_i) \in O\}$.

V_o: idle time after operation $o \in O$.

R_o: energy consumption during V_o.

$W_{o,q}$: 1 if $o \in O$ immediately precedes $q \in O$ on the same machine and 0 otherwise.

$Z_{o,k}$: 1 if turning shutdown strategy is implemented during V_o on machine $k \in K_o$ and 0 otherwise.

$Q_{o,k}$: 1 if $o \in O$ is processed last on machine $k \in K_o$ and 0 otherwise.

3.1.1. MILP-3 for Energy-Aware Flexible Job Shop Scheduling.

$$\text{Min} \sum_{o \in O} \sum_{k \in K_o} E_{o,k} X_{o,k} + \sum_{o \in O} R_o + P_0 C_{\max}, \tag{1}$$

$$\sum_{k \in K_o} X_{o,k} = 1, \quad \forall o \in O, \tag{2}$$

$$C_o = S_o + \sum_{k \in K_o} p_{o,k} X_{o,k}, \quad \forall o \in O, \tag{3}$$

$$S_q \geq C_o, \quad \forall (o, q) \in PO, \tag{4}$$

$$C_{\max} \geq C_o, \quad \forall o = (i, n_i) \in O, \tag{5}$$

$$\sum_{o \in O} Q_{o,k} \leq 1, \quad \forall k \in K, \tag{6}$$

$$\sum_{o \in O_q} W_{o,q} \leq 1, \quad \forall q \in O, \tag{7}$$

$$\sum_{q \in O_o} W_{o,q} + \sum_{k \in K_o} Q_{o,k} = 1, \quad \forall o \in O, \tag{8}$$

$$W_{o,q} - 1 \leq X_{o,k} - X_{q,k}, \quad \forall o \in O,\ q \in O_o,\ k \in K_o \cap K_q, \tag{9}$$

$$X_{o,k} - X_{q,k} \leq 1 - W_{o,q}, \quad \forall o \in O,\ q \in O_o,\ k \in K_o \cap K_q, \tag{10}$$

$$\sum_{k \in K_o \cap K_q} X_{o,k} \geq W_{o,q}, \quad \forall o \in O,\ q \in O_o, \tag{11}$$

$$\sum_{k \in K_o \cap K_q} X_{q,k} \geq W_{o,q}, \quad \forall o \in O,\ q \in O_o, \tag{12}$$

$$Q_{o,k} \leq X_{o,k}, \quad \forall o \in O,\ k \in K_o, \tag{13}$$

$$S_q \geq C_o + M_1 \left(W_{o,q} - 1 \right), \quad \forall o \in O,\ q \in O_o, \tag{14}$$

$$V_o \leq M_1 \left(1 - \sum_{k \in K_o} Q_{o,k} \right), \quad \forall o \in O, \tag{15}$$

$$V_o \leq S_q - C_o + M_1 \left(1 - W_{o,q} \right), \quad \forall o \in O,\ q \in O_o, \tag{16}$$

$$V_o \geq S_q - C_o - M_1 \left(1 - W_{o,q} \right), \quad \forall o \in O,\ q \in O_o, \tag{17}$$

$$\sum_{o \in O} Z_{o,k} \leq N_k, \quad \forall k \in K, \tag{18}$$

$$V_o \geq TB_k Z_{o,k}, \quad \forall o \in O,\ k \in K_o, \tag{19}$$

$$R_o \geq G_k Z_{o,k}, \quad \forall o \in O,\ k \in K_o, \tag{20}$$

$$R_o \geq D_k V_o + M_2 \left(X_{o,k} - Z_{o,k} - 1 \right), \quad \forall o \in O,\ k \in K_o, \tag{21}$$

$$Z_{o,k} \leq X_{o,k}, \quad \forall o \in O,\ k \in K_o, \tag{22}$$

$$Z_{o,k} \leq 1 - Q_{o,k}, \quad \forall o \in O,\ k \in K_o. \tag{23}$$

Objective (1) is to minimize the total energy, which is consumed for production, idle (including shutdown), and common. Constraint (2) ensures that each operation is processed by exactly one machine. Constraint (3) links the completion time of an operation with its starting time. Constraint (4) enforces that each operation starts after its precedent operation is finished. Constraint (5) indicates the makespan. Constraint (6) imposes that each machine has at most one last job. Constraint (7) imposes that each operation has at most one immediate precedence operation. Constraint (8) imposes that each operation has an immediate next operation or it is the last operation on some machine. Constraints (9)–(12) encode that if o immediately precedes q, then they should be processed by the same machine. Constraint (13) imposes that o can be the last job of machine k only when machine k processes o. Constraint (14) enforces that operation q starts after the completion of operation o if it is processed immediately after operation o on the same machine. Constraint (15) imposes that idle time of the last job is zero. Constraints (16)–(17) compute the idle time V_o of o. Constraint (18) restricts the number of times of shutdown strategy in machine k. Constraint (19) restricts the minimum length of V_o when shutdown strategy is used during V_o. Constraints (20)–(21) compute the idle energy consumption during V_o. Constraints (22)–(23) enforce that shutdown strategy can be turned during V_o on machine k only when o is processed by machine k and it is not the last operation of machine k. We term this newly proposed model MILP-3.

3.2. Constraint Programming. All time durations, such as makespan, shutdowns, and productions, are explicitly modeled as the interval variables as shown in Figure 2.

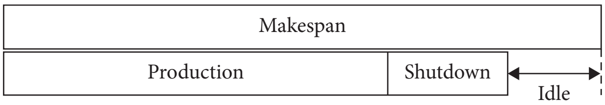


Figure 2: Modeling overview.

As Ham [41] pointed out, there is no standard in CP formulation, unlike a similar MILP formulation. Therefore, this paper formulates the model using generic keywords and syntaxes as we refer to the CP formulations by Laborie et al. [40] and IBM ILOG CP Optimizer [42].

The proposed CP model is built upon the following decision variables:

$Z_{k,s}$: interval representing s-th shutdown on machine $k \in K$

C_k : interval representing makespan of $k \in K$

T_o : interval representing operation $o \in O$

$X_{o,k}$: interval representing operation $o \in O$ on machine $k \in K_o$

$\mathrm{Seq}_k \leftarrow [X_{o,k}]$: collection of variables assigned to machine k

eS: total energy consumption during shutdowns

eP: total energy consumption during productions

eI: total energy consumption during idles

eC: total common energy consumption

3.2.1. CP-1.

$$\mathrm{Min}\, eP + eS + eI + eC, \tag{24}$$

$$\mathrm{alternative}\left(T_o, \left[X_{o,k:\, k\in K_o}\right]\right), \quad \forall o \in O, \tag{25}$$

$$\mathrm{endBeforeStart}\left(T_o, T_q\right), \quad \forall (o,q) \in PO, \tag{26}$$

$$\mathrm{noOverlap}\left(\mathrm{Seq}_k\right), \quad \forall k \in K, \tag{27}$$

$$\mathrm{span}\left(C_k, \left[X_{o,k:\, o\in O, k\in K}\right]\right), \quad \forall k \in K, \tag{28}$$

$$\mathrm{startOf}\left(Z_{k,s}\right) > \mathrm{startOf}\left(C_k\right), \quad \forall k \in K,\ s \le N_k, \tag{29}$$

$$\mathrm{endOf}\left(Z_{k,s}\right) < \mathrm{endOf}\left(C_k\right), \quad \forall k \in K,\ s \le N_k, \tag{30}$$

$$\mathrm{sizeOf}\left(Z_{k,s}\right) \ge TB_k, \quad \forall k \in K,\ s \le N_k, \tag{31}$$

$$eP = \sum_{o\in O} \sum_{k\in K_o} E_{o,k}\, \mathrm{sizeOf}\left(T_o\right), \tag{32}$$

$$eS = G_k \sum_{k\in K_o} \sum_{s \le N_k} \mathrm{presenceOf}\left(Z_{k,s}\right), \tag{33}$$

$$eI = D_k \sum_{k\in K} \left(\mathrm{sizeOf}\left(C_k\right) - \sum_{s \le N_k} \mathrm{sizeOf}\left(Z_{k,s}\right) - \sum_{o\in O} \mathrm{sizeOf}\left(X_{o,k}\right) \right), \tag{34}$$

$$eC = P_0 \max_{\forall o\in O} \left\{ \mathrm{endOf}\left(T_o\right) \right\}. \tag{35}$$

Objective (24) is to minimize the total energy consumption, which is comprised of production, shutdown, idle, and common. Constraint (25) ensures that each operation is processed by exactly one machine. Constraint (26) enforces each operation starts after its precedent operation is finished. Constraint (27) prevents intervals in a sequence from overlapping. Constraint (28) determines the makespan of each machine. Constraints (29)–(30) ensure that a shutdown can occur during a makespan. Constraint (31) ensures the minimum length of each shutdown. Constraints (32)–(35) compute the total energy consumption for production, shutdown, idle, and common, respectively. We term this model CP-1.

During the preliminary study, the CP-1 could not outperform the MILP-2A model. Here, we explored an alternative CP model. In the CP-1, the idle time was computed based on other interval variables in Constraint (34). We here present an alternative CP model, which explicitly captures the idle intervals.

$\mathbf{I}_{k,t}$: interval representing t-th idle on machine $k \in K$.

$$eI = D_k \sum_{k\in K} \sum_{t \le m_k} \text{sizeOf}\left(I_{k,t}\right), \quad (36)$$

$$\text{sizeOf}\left(C_k\right) = \sum_{o\in O} \text{sizeOf}\left(X_{o,k}\right) + \sum_{s \le N_k} \text{sizeOf}\left(Z_{k,s}\right) + \sum_{t \le m_k} \text{sizeOf}\left(I_{k,t}\right). \quad (37)$$

Constraint (36) replaces Constraint (34). Then, Constraint (37) is introduced to ensure that a makespan of the machine must be packed by productions, shutdowns, and idles. All other constraints stay in place. We term this revised model CP-2.

Figure 3 represents an optimal schedule for a small benchmark instance (mfjs07). Figure 3(a) shows the detailed production schedule per machine with an objective of makespan minimization, and Figure 3(b) depicts the schedule with an objective of energy minimization, while maintaining the same productivity. The proposed model merges the small-size idles enough to turn on shutdown strategy. For instance, the two idles that occurred in machine 4 are replaced by a single shutdown in the same machine. The energy-aware scheduling saved the energy consumption by 1.35% without hurting productivity in the mfjs07 instance.

4. Computational Experiments

In this section, the effectiveness of the proposed models is examined. The MILP, CP, and flow control models are all coded in IBM OPL 12.8.0 on a personal computer with an Intel® Core i7-4770 CPU with 16 GB of RAM. All the test instances, MILP codes, CP codes in IBM OPL, and GA code can be found online at https://github.com/hamcruise/FJSP-Shutdown.

4.1. Problem Instances. Meng et al. [9] proposed a set of test instances derived from the known benchmark instances SFJS01-10 and MFJS01-10 [12] by considering energy consumption. We adopted the same test instances. SFJS01-10 are small-sized, and MFJS01-10 are medium-sized. We also adopted another FJSP benchmark test instance suggested by Behnke and Geiger [43]. Among their extensive 60 instances, we adopted the first 10 instances and added energy parameters in the same way Meng et al. [9] did. Finally, we also used another FJSP benchmark test instance suggested by Kacem et al. [44], which has been adopted by Singh and Mahapatra [45]. The detailed size of the instance (*a*/*b*/*c*) is recorded in Table 2, where index *a* denotes the number of jobs, *b* denotes the maximum number of operations for a job, and *c* denotes the number of machines.

For all instances, the common power (P_0) and maximum times of shutdown (N_k) are set to be 5 and 3, respectively. The processing powers are drawn from the uniform distribution [3, 5]. The idle power (D_k) is randomly generated from the set {1, 2, 3}, and the energy consumption during shutdown (G_k) is generated from the set {10, 30, 60}.

4.2. Experimental Results. Figure 4 shows the ratio of the number of binary variables of MILP models to MILP-2. As the size of MILP-2 increases, the binary variables of MILP-2A and MILP-3 decrease. In particular, it is observed that the reduction ratio of the binary variable of MILP-3 sharply decreases as the size increases.

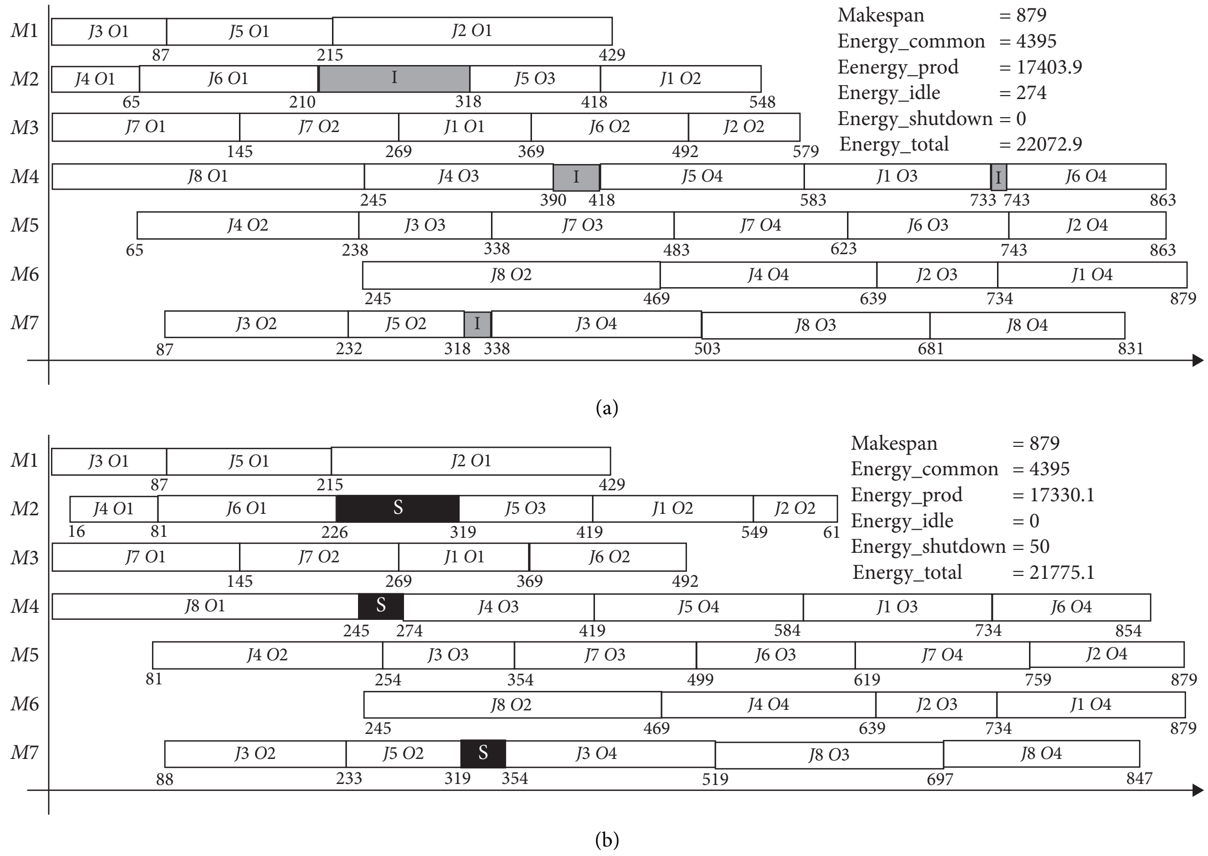
In this subsection, we implemented a genetic algorithm to compare the performance between proposed methods and metaheuristic. Genetic algorithm (GA) is known as an effective metaheuristic to solve flexible job shop scheduling problems [46–49]. The overall GA framework in Zhang et al. [50] was adopted. We had conducted a preliminary test for tuning hyperparameters with respect to the quadruple (*popSize, numGen, crRate*, *mutRate*), where *popSize* is the population size (*popSize* ∈ {100, 200, 400}), *numGen* is the maximum number of generations (*numGen* ∈ {50, 100, 200}), *crRate* is the crossover rate (*crRate* ∈ {0.6, 0.7, 0.8, 0.9, 1.0}), and *mutRate* is the mutation rate (*mutRate* ∈ {0.0, 0.1, 0.2, 0.3}). The best GA hyperparameters that were obtained from the preliminary test were as follows: *popSize* = 100, *numGen* = 50, *crRate* = 0.7, and *mutRate* = 0.1. The increasing population size and number of generations did not ensure obtaining better solutions. The value of the fitness function converged fast in practice.

Tables 3 and 4 compare the proposed MILP and GA with CP models in terms of the objective function value (total energy consumption) and computation times within 600 seconds for Fattahi and Behnke's instances, respectively. Column 1 identifies the name of the instance, columns 2–7 include the total energy consumption, and columns 8–13 report the computation times. The bold font indicates the optimality.

Table 3 reports the results based on Fattahi's instances. For small-sized instances (SFJS01-10), all approaches found the optimal solutions, except GA. In computation times, all approaches, except CP-1, terminated in one second. For medium-sized instances (MFJS01-10), CP-2 yielded the best results on average. In computation times, GA significantly outperforms all other approaches seconds. However, MILP-2, MILP-2A, and MILP-3 terminated at a similar time.

Table 4 reports the results based on Behnke's instances. The proposed MILP-2A performed the best in average. GA is significantly faster than the MILP and CP approaches. However, the total energy consumption is increased up to 8.1% on average.

In order to determine if the means of two sets of data are significantly different from each other, we conducted the *t*-test: Paired Two-Sample for Means at an alpha level of 0.05. Table 5 indicates that, for medium-sized Fattahi's instances, there is a statistically significant difference between CP-2 vs. CP-1 and GA, while there is no statistically significant difference between CP-2 vs. MILP-2, MILP-2A, and MILP-3. For Behnke's instances, there is a statistically significant difference between MILP-2A vs. CP-1, MILP-2, MILP-2A,



(a)

(b)

Figure 3: Schedules for the mfjs07 instance. (a) Traditional scheduling to minimize $C_{\max}$. (b) Energy-aware scheduling to minimize energy consumption while keeping the same productivity.

Table 2: Characteristics of the test instances.

Fattahi instances				Behnke instances	
Name	Size	Name	Size	Name	Size
sfjs01	2/2/2	mfjs01	5/3/6	Behnke1	10/3/20
sfjs02	2/2/2	mfjs02	5/3/7	Behnke2	10/3/20
sfjs03	3/2/2	mfjs03	6/3/7	Behnke3	10/3/20
sfjs04	3/2/2	mfjs04	7/3/7	Behnke4	10/3/20
sfjs05	3/2/2	mfjs05	7/3/7	Behnke5	10/3/20
sfjs06	3/3/2	mfjs06	8/3/7	Behnke6	20/3/20
sfjs07	3/3/5	mfjs07	8/4/7	Behnke7	20/3/20
sfjs08	3/3/4	mfjs08	9/4/8	Behnke8	20/3/20
sfjs09	3/3/3	mfjs09	11/4/8	Behnke9	20/3/20
sfjs10	4/3/5	mfjs10	12/4/8	Behnke10	20/3/20
		Kacem instances			
Name	Size	Name	Size	Name	Size
Kacem1	8/4/8	Kacem2	10/3/10	Kacem3	15/4/10

MILP-3, and GA methods, while there is no statistically significant difference between MILP-2A and CP-2.

The previous tables compared the performance of models with a fixed computation time. It is necessary to check how the performance changes according to different computation times. Table 6 summarizes the total energy consumption in terms of computational run time for the Kacem's instances. Both CP and MILP models proved the optimality of Kacem1 instance within 60 s. CP managed to prove the optimality of Kacem2 instance within 60 s, while MILP spent 150 s. In the Kacem3 instance, the CP-2 found an optimal solution within 60 s, while MILP-2A could not find an optimal solution within 600 s. In addition, the CP-2 found a much better solution, in just 10 seconds, than the one MILP-2A found in 600 seconds. This experiment demonstrates that CP is quick to generate efficient (or optimal) solutions.

The energy saving does not appeal to highly capitalized manufacturing industries, such as automobiles and

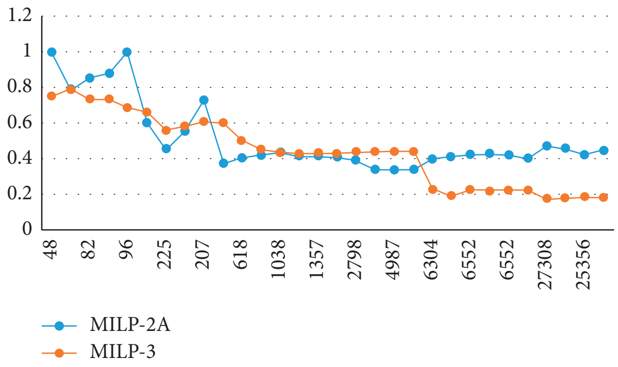


Figure 4: Comparison of the number of binary variables of MILP models.

Table 3: Comparison of models in terms of total cost and computation times on Fattahi instances.

Instance	Total energy consumption						Computation time (second)					
	CP-1	CP-2	MILP-2	MILP-2A	MILP-3	GA	CP-1	CP-2	MILP-2	MILP-2A	MILP-3	GA
sfjs01	815.2	815.2	815.2	815.2	815.2	815.2	54.6	0.3	0.0	0.0	0.0	0.2
sfjs02	1362.2	1362.2	1362.2	1362.2	1362.2	1362.2	65.4	0.3	0.0	0.0	0.0	0.2
sfjs03	2806.2	2806.2	2806.2	2806.2	2806.2	2806.2	507.3	0.3	0.0	0.0	0.0	0.2
sfjs04	4560.3	4560.3	4560.3	4560.3	4560.3	4560.3	600.1	0.3	0.0	0.0	0.0	0.2
sfjs05	1405.4	1405.4	1405.4	1405.4	1405.4	1405.4	547.7	0.3	0.0	0.0	0.0	0.2
sfjs06	4304.6	4304.6	4304.6	4304.6	4304.6	4420.6	600.1	0.3	0.1	0.0	0.0	0.3
sfjs07	5256.0	5256.0	5256.0	5256.0	5256.0	5345.2	600.0	0.3	0.1	0.0	0.0	0.3
sfjs08	3429.7	3429.7	3429.7	3429.7	3429.7	3477.7	600.0	0.3	0.3	0.1	0.0	0.3
sfjs09	2848.0	2848.0	2848.0	2848.0	2848.0	3023.2	600.2	0.4	0.1	0.1	0.1	0.3
sfjs10	8877.0	8877.0	8877.0	8877.0	8877.0	8877	600.0	0.4	0.1	0.0	0.0	0.4
mfjs01	9380.7	9380.7	9380.7	9380.7	9380.7	9981.2	600.0	600.1	2.9	0.8	1.6	0.5
mfjs02	8642.0	8642.0	8642.0	8642.0	8642.0	9082	600.0	600.0	2.8	1.1	1.9	0.5
mfjs03	10799.4	10762.8	10757.8	10757.8	10757.8	11700	600.0	600.0	35.3	4.7	10.8	0.6
mfjs04	13075.9	13038.6	13038.6	13038.6	13038.6	13696	600.0	600.1	117.5	27.3	82.3	0.6
mfjs05	12744.4	12600.1	12600.1	12600.1	12600.1	13579.1	600.0	600.1	93.3	23.7	426.8	0.6
mfjs06	15169.7	14960.1	14960.1	14960.1	14960.1	16257.5	600.0	600.1	49.6	16.7	52.0	0.7
mfjs07	20842.1	20542.1	21098.3	20990.3	21179.6	21884.9	600.0	600.0	607.5	600.2	600.3	1.0
mfjs08	24295.0	23763.8	24289.6	23999.0	23993.4	25812.3	600.0	600.1	601.8	600.3	600.3	1.0
mfjs09	30999.4	29788.1	31386.7	30882.5	30846.4	32237.1	600.1	600.0	601.1	600.5	600.1	1.2
mfjs10	34946.8	34410.7	36794.5	35168.6	37136.6	37293.6	600.0	600.1	600.4	601.6	600.8	1.5
Average	10828.0	10677.7	10930.7	10804.2	10910.0	11380.8	538.8	300.2	135.7	123.9	148.9	0.5

Table 4: Comparison of models in terms of total cost and computation times on Behnke instances.

Instance	Total energy consumption						Computation time (second)					
	CP-1	CP-2	MILP-2	MILP-2A	MILP-3	GA	CP-1	CP-2	MILP-2	MILP-2A	MILP-3	GA
Behnke1	1839.2	1795.8	1795.8	1795.8	1795.8	1889.3	600.0	600.1	600.1	104.6	600.1	1.1
Behnke2	1787.8	1768.0	1763.9	1763.9	1763.9	1884.6	600.0	600.1	318.3	167.3	180.8	1.1
Behnke3	1752.4	1749.9	1749.9	1749.9	1749.9	1902.6	600.0	600.2	332.6	55.3	309.2	1.1
Behnke4	2012.4	1964.2	1961.4	1945.9	1950.4	2082.9	600.0	600.1	600.2	379.7	600.1	1.1
Behnke5	1928.2	1865.7	1867.3	1865.7	1872.6	2101.1	600.0	600.1	600.3	124.4	600.1	1.0
Behnke6	3565.1	3354.1	3385.8	3361.3	3381.6	3637.7	600.1	600.2	600.4	600.2	600.1	2.2
Behnke7	3875.4	3483.1	3475.3	3454.6	3455.2	3734.9	600.0	600.2	602.7	600.3	600.3	2.2
Behnke8	3701.7	3445.2	3509.6	3474.4	3521.3	3774.9	600.0	600.3	604.4	600.4	600.3	1.9
Behnke9	3526.5	3326.6	3324.4	3294.4	3340.0	3579.2	600.0	600.3	600.4	600.3	600.2	2.1
Behnke10	4150.0	3703.4	3742.8	3705.7	3784.6	3916.8	600.0	600.3	603.3	600.4	600.3	2.1
Average	2813.9	2645.6	2657.6	2641.2	2661.5	2850.4	600.0	600.2	546.3	383.3	529.1	1.6

semiconductors, because the total energy cost is not even comparable to the revenue the manufacturers yield, so their sole goal is to maximize productivity. However, if we can maintain the same productivity, while significantly reducing energy consumption, the energy-aware production scheduling will be embraced by the industries. This type of

Table 5: Comparison of models in terms of *t*-test: paired two-sample for means.

CP-2 vs.	Medium-sized Fattahi instances (MFJS01-10)					MILP-2A vs.	Behnke instances				
	CP-1	MILP-2	MILP-2A	MILP-3	GA		CP-1	CP-2	MILP-2	MILP-3	GA
t-value *t* (9)	−2.511	−1.918	−2.038	−1.684	−5.185	*t*-value *t* (9)	−3.387	−0.780	−3.426	−2.337	−8.547
p-value	0.0333	0.0874	0.0720	0.1265	0.0006	*p*-value	0.0080	0.4552	0.0076	0.0442	0.0000

Table 6: Comparison of models in terms of total energy consumption according to different CPU times.

Instance	Model	CPU (sec)								
		10	30	60	90	120	150	180	300	600
Kacem1	CP-2	407.5	401.4	393.4	—	—	—	—	—	—
	MILP-2A	443.0	408.6	393.4	—	—	—	—	—	—
	GA	422.6								
Kacem2	CP-2	205.8	202.2	200.8	—	—	—	—	—	—
	MILP-2A	238.6	220.8	215.5	210.0	201.5	200.8	—	—	—
	GA	227.5								
Kacem3	CP-2	520.2	435.8	435.8	435.8	435.8	435.8	435.8	435.8	435.8
	MILP-2A	1909.2	1908.1	1908.1	1908.1	1908.1	1908.1	1908.1	1908.1	1908.1
	GA	501.8								

Table 7: Potential energy saving based on Fattahi instances.

Instance	Total energy consumption			Energy saving	
	Tradition	Energy-aware scheduling		Energy-aware scheduling	
		w/o shutdown	w/ shutdown	w/o shutdown (%)	w/ shutdown (%)
sfjs01	815.2	815.2	815.2	0.00	0.00
sfjs02	1362.2	1362.2	1362.2	0.00	0.00
sfjs03	2806.2	2806.2	2806.2	0.00	0.00
sfjs04	4560.3	4560.3	4560.3	0.00	0.00
sfjs05	1405.4	1405.4	1405.4	0.00	0.00
sfjs06	4630.6	4360.6	4360.6	5.83	5.83
sfjs07	5530.2	5304.2	5304.2	4.09	4.09
sfjs08	3599.2	3599.2	3599.2	0.00	0.00
sfjs09	3121.0	2951.0	2951.0	5.45	5.45
sfjs10	9217.8	8893.0	8877.0	3.52	3.70
mfjs01	9498.6	9468.6	9468.6	0.32	0.32
mfjs02	9036.2	8918.2	8918.2	1.31	1.31
mfjs03	11410.0	11356.0	11278.0	0.47	1.16
mfjs04	13233.9	13163.9	13075.9	0.53	1.19
mfjs05	13333.6	13293.6	13293.6	0.30	0.30
mfjs06	16443.1	16086.7	16086.7	2.17	2.17
mfjs07	22072.9	21973.1	21775.1	0.45	1.35
mfjs08	24884.4	24509.6	24503.8	1.51	1.53
mfjs09	32144.4	31262.2	31262.2	2.74	2.74
mfjs10	35857.7	34874.9	34874.9	2.74	2.74
		Average		1.57	1.69

mathematical approach is known as epsilon-constraint method, where one of the objectives is taken as a single objective function and the others are included into a model as constraint [51].

Tables 7 and 8 attempt to meet this demand by comparing the traditional scheduling and energy-aware scheduling. The traditional scheduling minimizes the makespan (cycle time reduction), whereas the energy-aware scheduling minimizes the total energy consumption as it maintains the same makespan. Column 1 identifies the name of the instance. Column 2 reports the total energy consumption when the model solves each test instance with the makespan minimization. Columns 3-4 report the total energy consumption when the model runs with the energy consumption minimization with the same makespan. This experimentation identifies potential energy-saving without compromising productivity. For Fattahi's instances, Table 7 shows 1.57% and 1.69% saving without and with shutdown

Table 8: Potential energy saving based on Behnke instances.

Instance	Tradition	Total energy consumption		Energy saving	
		Energy-aware scheduling		Instance	
		w/o shutdown	w/ shutdown	w/o shutdown (%)	w/ shutdown (%)
Behnke1	2156.2	1809.9	1809.9	16.06	16.06
Behnke2	1955.2	1789.3	1789.3	8.49	8.49
Behnke3	2106.1	1840.9	1796.3	12.59	14.71
Behnke4	2237.8	1945.9	1945.9	13.04	13.04
Behnke5	2125.1	1953.2	1953.2	8.09	8.09
Behnke6	4356.7	3358.3	3358.3	22.92	22.92
Behnke7	4210.8	3543.7	3543.7	15.84	15.84
Behnke8	4200.7	3502.8	3502.8	16.61	16.61
Behnke9	4022.5	3341.8	3297.9	16.92	18.01
Behnke10	4558.5	3740.1	3714.5	17.95	18.51
Average				14.85	15.23

Table 9: Energy saving according to different energy consumption.

Instance	$E_{o,k}$ (%)	$2E_{o,k}$ (%)	$3E_{o,k}$ (%)	$4E_{o,k}$ (%)
Kacem1	10.5	9.5	9.2	8.9
Kacem2	2.8	1.7	1.3	1.1
Kacem3	11.4	9.7	8.9	8.6
Average	8.2	7.0	6.5	6.2

strategy, respectively. The mild saving can be explained by the test instance itself.

Table 8 reports the results of Behnke's instances. This experimentation identified the potential energy-saving without compromising productivity by 14.85% (w/o shutdown) and 15.23% (w/shutdown). The increased savings compared with the ones with Fattahi's instance are due to the size of this Behnke's instance, allowing the solver the flexibility to find better solutions. In order to determine if the means of two sets of data are significantly different from each other, we conducted the t-test: Paired Two-Sample for Means at an alpha level of 0.05. The test indicates that there is a statistically significant difference between traditional and energy-aware methods with $t\ (19) = 3.5$ and $p < 0.01$ in Fattahi's instance and $t\ (9) = 5.53$ and $p < 0.001$ in Behnke's instance.

Table 9 records the amount of energy saving by applying the proposed energy-aware method according to different multipliers on $E_{o,k}$. This experimentation identified the potential energy-saving without compromising productivity by 6–8%. The improvement was slightly decreased as the multiplier was increased.

5. Conclusion

We have investigated the energy-efficient FJSP with a shutdown (on/off) strategy to save idle energy consumption. The shutdown can be enforced when a continuous idle period of a machine is expected to be long enough to compensate for the shutdown penalty. An alternative MILP model is proposed. Then, a novel constraint programming is proposed. Finally, practical operational scenarios are examined.

The computational study demonstrated that (1) the proposed models significantly outperform the best benchmark model (MILP-2) and (2) we can maintain the maximum productivity while significantly reducing the energy-consumption by 14.85% (w/o shutdown) and 15.23% (w/ shutdown) on average, thus promoting energy-aware production scheduling to highly capitalized manufacturing industries. We offer benchmarking instances, CPLEX MIP, CP, and GA source codes, which have been used in this research, in order to promote related research, thus expediting the adoption of energy-efficient scheduling in manufacturing facilities.

In future research, we will extend the proposed models to consider time-of-use (TOU) electricity and peak power load since the energy cost can be further saved. Utility companies across the U.S. are offering TOU-based electricity demand response programs. The key is to shift productions to off-peak periods.

Data Availability

The data used to support the findings of this study have been deposited in the GitHub repository (https://github.com/hamcruise/FJSP-Shutdown)

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (no. 2019R1G1A1085191).

References

[1] "International energy outlook 2019 with projections to 2050," 2019, https://www.eia.gov/ieo.

[2] International Energy Agency, *Tracking Industrial Energy Efficiency and CO2 Emissions*, Organisation for Economic Co-Operation and Development, Paris, France, 2007.

[3] G. May, B. Stahl, M. Taisch, and V. Prabhu, “Multi-objective genetic algorithm for energy-efficient job shop scheduling,” *International Journal of Production Research*, vol. 53, no. 23, pp. 7071–7089, 2015.

[4] A. M. Stark, *U.S. Energy Use Rises to Highest Level Ever*, Lawrence Livermore National Laboratory (LLNL) News Release, Livermore, CA, USA, 2019.

[5] G. Mouzon, M. B. Yildirim, and J. Twomey, “Operational methods for minimization of energy consumption of manufacturing equipment,” *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[6] Y. He, F. Liu, H.-J. Cao, and C.-B. Li, “A bi-objective model for job-shop scheduling problem to minimize both energy consumption and makespan,” *Journal of Central South University of Technology*, vol. 12, no. 2, pp. 167–171, 2005.

[7] A. Che, X. Wu, J. Peng et al., “Energy-efficient bi-objective single-machine scheduling with power-down mechanism,” *Computers & Opereations Research*, vol. 85, pp. 172–183, 2017.

[8] L. Zhang, Q. Tang, Z. Wu, and F. Wang, “Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops,” *Energy*, vol. 138, pp. 210–227, 2017.

[9] L. Meng, C. Zhang, X. Shao, and Y. Ren, “MILP models for energy-aware flexible job shop scheduling problem,” *Journal of Cleaner Production*, vol. 210, pp. 710–723, 2019.

[10] I. A. Chaudhry and A. A. Khan, “A research survey: review of flexible job shop scheduling techniques,” *International Transactions in Operational Research*, vol. 23, no. 3, pp. 551–591, 2016.

[11] H. M. Wagner, “An integer linear-programming model for machine scheduling,” *Naval Research Logistics Quarterly*, vol. 6, no. 2, pp. 131–140, 1959.

[12] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, “Mathematical modeling and heuristic approaches to flexible job shop scheduling problems,” *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 331–342, 2007.

[13] N. Imanipour, “Modeling & solving flexible job shop problem with sequence dependent setup times,” in *Proceedings of the International Conference on Service Systems and Service Management*, vol. 2, pp. 1205–1210, Troyes, France, October 2006.

[14] C. Özgüven, L. Özbakır, and Y. Yavuz, “Mathematical models for job-shop scheduling problems with routing and process plan flexibility,” *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1539–1548, 2010.

[15] I.-C. Choi and D.-S. Choi, “A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups,” *Computers & Industrial Engineering*, vol. 42, no. 1, pp. 43–58, 2002.

[16] M. C. Gomes, A. P. Barbosa-Póvoa, and A. Q. Novais, “Optimal scheduling for flexible job shop operation,” *International Journal of Production Research*, vol. 43, no. 11, pp. 2323–2353, 2005.

[17] Y. Demir and S. Kürşat İşleyen, “Evaluation of mathematical models for flexible job-shop scheduling problems,” *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 977–988, 2013.

[18] Q. Jiang, X. Liao, R. Zhang, and Q. Lin, “Energy-saving production scheduling in a single-machine manufacturing system by improved particle swarm optimization,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 8870917, 16 pages, 2020.

[19] J.-Q. Li, H.-Y. Sang, Y.-Y. Han, C.-G. Wang, and K.-Z. Gao, “Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions,” *Journal of Cleaner Production*, vol. 181, pp. 584–598, 2018.

[20] B. Zhang, Q. K. Pan, L. Gao, L. L. Meng, X. Y. Li, and K. K. Peng, “A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 4984–4999, 2019.

[21] T. Jiang, C. Zhang, H. Zhu, and G. Deng, “Energy-efficient scheduling for a job shop using grey wolf optimization algorithm with double-searching mode,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 8574892, 12 pages, 2018.

[22] L. Yin, X. Li, L. Gao, C. Lu, and Z. Zhang, “A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem,” *Sustainable Computing: Informatics and Systems*, vol. 13, pp. 15–30, 2017.

[23] Z. Jiang, L. Zuo, and E. Mingcheng, “Study on multi-objective flexible job-shop scheduling problem considering energy consumption,” *Journal of Industrial and Engineering and Management*, vol. 7, no. 3, pp. 589–604, 2014.

[24] L. Meng, C. Zhang, X. Shao, Y. Ren, and C. Ren, “Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines,” *International Journal of Production Research*, vol. 57, no. 4, pp. 1119–1145, 2019.

[25] M. Dai, D. Tang, A. Giret, M. A. Salido, and W. D. Li, “Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 418–429, 2013.

[26] X. Wu and Y. Sun, “A green scheduling algorithm for flexible job shop with energy-saving measures,” *Journal of Cleaner Production*, vol. 172, pp. 3249–3264, 2018.

[27] L. Meng, C. Zhang, B. Zhang, and Y. Ren, “Mathematical modeling and optimization of energy-conscious flexible job shop scheduling problem with worker flexibility,” *IEEE Access*, vol. 7, pp. 68043–68059, 2019.

[28] A. Che, K. Lv, E. Levner, and V. Kats, “Energy consumption minimization for single machine scheduling with bounded maximum tardiness,” in *Proceedings of the ICNSC 2015-2015 IEEE 12th International Conference of Networking*, pp. 146–150, Taipei, Taiwan, April 2015.

[29] K. Fang, N. A. Uhan, F. Zhao, and J. W. Sutherland, “Flow shop scheduling with peak power consumption constraints,” *Annals of Operations Research*, vol. 206, no. 1, pp. 115–145, 2013.

[30] S. A. Mansouri, E. Aktas, and U. Besikci, “Green scheduling of a two-machine flowshop: trade-off between makespan and energy consumption,” *European Journal of Operational Research*, vol. 248, no. 3, pp. 772–788, 2016.

[31] R. Zhang and R. Chiong, “Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption,” *Journal of Cleaner Production*, vol. 112, pp. 3361–3375, 2016.

[32] L. Zhang, X. Li, L. Gao, G. Zhang, and X. Wen, “Dynamic scheduling model in FMS by considering energy consumption and schedule efficiency,” in *Proceedings of the IEEE International Conference on Computer Supported Cooperative Work in Design*, Wuhan, China, May 2012.

[33] M. Rager, C. Gahm, and F. Denz, “Energy-oriented scheduling based on evolutionary algorithms,” *Computers & Operations Research*, vol. 54, pp. 218–231, 2015.

[34] S. Schulz, J. S. Neufeld, and U. Buscher, "A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling," *Journal of Cleaner Production*, vol. 224, pp. 421–434, 2019.

[35] J.-Y. Moon and J. Park, "Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed energy resources and energy storage," *International Journal of Production Research*, vol. 52, no. 13, pp. 3922–3939, 2014.

[36] X. Gong, T. De Pessemier, L. Martens, and W. Joseph, "Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: a many-objective optimization investigation," *Journal of Cleaner Production*, vol. 209, pp. 1078–1094, 2019.

[37] C. Gahm, F. Denz, M. Dirr, and A. Tuma, "Energy-efficient scheduling in manufacturing companies: a review and research framework," *European Journal of Operational Research*, vol. 248, no. 3, pp. 744–757, 2016.

[38] K. Gao, Y. Huang, A. Sadollah, and L. Wang, "A review of energy-efficient scheduling in intelligent production systems," *Complex & Intelligent Systems*, vol. 6, pp. 1–13, 2019.

[39] F. S. Hiller and G. J. Lieberman, *Introduction to Operations Research*, Prentice Hall, Hoboken, NJ, USA, 10th edition, 2015.

[40] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím, "IBM ILOG CP optimizer for scheduling," *Constraints*, vol. 23, no. 2, pp. 210–250, 2018.

[41] A. Ham, "Transfer-robot task scheduling in job shop," *International Journal of Production Research*, vol. 59, no. 3, pp. 813–823, 2020.

[42] IBM Software, *IBM ILOG CPLEX Optimization Studio V12.8.0*, IBM, Endicott, NY, USA, 2019.

[43] D. Behnke and M. J. Geiger, *Test Instances For The Flexible Job Shop Scheduling Problem With Work Centers*, Helmut-Schmidt-Universität, Hamburg, Germany, 2012.

[44] I. Kacem, S. Hammadi, and P. Borne, "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)*, vol. 32, no. 1, pp. 1–13, 2002.

[45] M. R. Singh and S. S. Mahapatra, "A quantum behaved particle swarm optimization for flexible job shop scheduling," *Computers & Industrial Engineering*, vol. 93, pp. 36–44, 2016.

[46] J. Gao, L. Sun, and M. Gen, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems," *Computers & Operations Research*, vol. 35, no. 9, pp. 2892–2907, 2008.

[47] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 904–916, 2019.

[48] M. A. González, A. Oddi, and R. Rasconi, "Efficient approaches for solving a multiobjective energy-aware job shop scheduling problem," *Fundamenta Informaticae*, vol. 167, no. 1-2, pp. 93–132, 2019.

[49] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3202–3212, 2008.

[50] G. Zhang, L. Gao, and Y. Shi, "An effective genetic algorithm for the flexible job-shop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3563–3573, 2011.

[51] H. G. Resat, "Design and analysis of novel hybrid multi-objective optimization approach for data-driven sustainable delivery systems," *IEEE Access*, vol. 8, pp. 90280–90293, 2020.