Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Novel Global Sequence-based Mathematical Formulation for Energy-efficient Flexible Job Shop Scheduling Problem

D. Li[a], and T.C. Zheng[a], and J. Li[a]*

[a] The University of Manchester, Department of Chemical Engineering, Manchester, UK
* Corresponding Author: jie.li-2@manchester.ac.uk

ABSTRACT

With increasing emphasis on energy efficiency, more researchers are focusing on energy-efficient flexible job shop scheduling problems. Mathematical programming is a commonly used optimization method for such scheduling challenges, offering the advantages of achieving global optima and serving as a foundation for other approaches. However, current mathematical programming formulations face several challenges, including insufficient consideration of various forms of energy consumption and low efficiency, particularly in handling large-scale instances, which struggle to converge. In this study, we propose a novel global sequence-based approach with high computational efficiency. In this model, immediate precedence relationships are identified using constraints, enabling the precise determination of idle durations within any idle slots. The proposed formulation achieves a significant reduction in energy consumption by up to 20% relative to other formulations. Furthermore, it successfully reaches feasible solutions in challenging cases.

Keywords: Algorithms, Energy Efficiency, Optimization, Scheduling, Energy

INTRODUCTION

The job-shop scheduling problem (JSSP) has been widely existing in various fields such as enterprise management [1,2], transportation [3] and aerospace [4,5]. As the demands for small-volume and personalized products increases, enterprises orientate at improving the flexibility of manufacturing processes by introducing parallel machines, which is generally termed as the flexible job-shop scheduling problem (FJSSP) [6]. In this content, assignment of operations to a machine as well as their sequence on the machine should be determined, making it more challenging to solve than JSSP. The inherent flexibility of FJSSP provides potential for significant economic or energy-efficient savings reflected in a schedule solution with the best assignments and sequences. Many approaches have been developed to address FJSSP, including exact algorithms [7-8], heuristics [9], and metaheuristics [10-11]. Several excellent reviews about the development, methodologies, and applications of FJSSP have been provided in the literature [12-13].

The prior research on FJSSP primarily centered around the economic performance or makespan, neglecting the simultaneous consideration of energy consumption. As manufacturing has the predominant contribution to annual industrial energy consumption, constituting the largest share at 76% among four major industry sectors in 2022 [14], the optimal schedule without consideration of energy consumption may lead to large energy wastage. Various forms of energy consumption have been identified, including standby, processing, turn-on/off, and setup energy consumption [8,15]. Notably, it was observed that 65% of the total energy consumption (TEC) in FJSSP is unrelated to actual processing activities. Instead, this substantial portion of energy is expended during periods when processing units are either idle or in standby mode [15]. The separation between economic optimization and energy awareness in production scheduling has significant implications. Production schedules optimized solely for economic goals may inadvertently result in unnecessarily high energy consumption. Consequently, there has been a growing recognition of the importance of incorporating energy considerations into FJSSP, especially due to net-zero strategies proposed by different countries.

Investigations have been conducted to propose optimization approaches for the energy-efficient FJSSP, which can be broadly categorized into exact methods and approximate methods. Among the exact methods, mathematical programming approaches, such as mixed-integer linear programming (MILP) formulations, are commonly utilized [16]. The MILP formulations excel in generating optimal solutions and serve as a foundation in understanding the characteristics of scheduling problems [17]. Additionally, the development of MILP formulations lays the groundwork for establishing decomposition approaches that are highly effective in handling the complexities of energy-efficient FJSSP.

Researchers [17-22] developed MILP models for energy-efficient FJSSP based on the time-grid representation. [18] and [19] discretized the time horizon into uniform time intervals and introduced binary variables to indicate whether an operation starts processing at a given time point. Unlike discrete-time representations, continuous-time representations divide the time horizon into events with unknown and variable durations. Zhang et al. [20] proposed an event-based MILP model to minimize the TEC, to establish comparison benchmarks for a gene-expression programming approach and to explicitly analyse the problem's features. Followingly, a more efficient event-based unit- specific model was proposed [21], incorporating auxiliary states to link adjacent operations within the same job and ensure their precedence constraints. These aforementioned studies comprehensively consider various forms of energy consumption, including standby energy, switch off/on energy, direct (i.e., processing) energy and indirect energy. However, these exhibit inefficiencies when applied to large-scale energy-efficient FJSSP systems. For example, they struggle to find feasible solutions within hours for problems involving more than 200 operations [20,21]. Further research on MILP models with continuous-time representations can be found in [17,22,23].

In addition to time-grid representation, MILP formulations can be developed based on the precedence relations between operations on machines. For example, [8] and [24] proposed the local-sequence-based formulation, where binary variables indicate whether an operation immediately precedes another on a machine. In contrast, the global-sequence-based formulation focuses on global precedence relations, requiring fewer binary variables, making it more efficient. This is because the binary variable denoting the first or last operations on machines are omitted. [25] presented a novel global sequence-based MILP formulation to optimize the minimization of makespan or TEC. However, they did not account for the selection between standby and switch off-on modes for machines, which is crucial for reducing energy waste during machine idle times. More specifically, their MILP model fails to accurately estimate the specific durations of each idle period on machines.

In this work, a global sequence-based MILP formulation (**MG**) is proposed to minimize TEC, encompassing direct, indirect and unload energy consumption. Its novelty lies in the incorporation of constraints that identify immediate precedence relations between operations and the optimal duration of idle slot between them. Selection between machine modes during idle slots is achieved by introducing a new set of binary variables. The proposed model is compared with MILP models based on time-grid representations and precedence relations as well as a decomposition approach.

PROBLEM DEFINITION

Fig. 1 illustrates an energy-efficient FJSSP involving a set of jobs $k \in \mathbf{K}$ and machines $j \in \mathbf{J}$. A job k is processed with a sequence of operations $i \in \mathbf{I}_k$. Machines j that can process operation i of job k are included in a set $\mathbf{J}_{ki}$. The processing time for machine j performing operation i of job k is denoted by P_{kij}^{T}, and the cutting power is P_{kij}^{C}. The objective is to minimize TEC, encompassing energy consumed by machines during processing operations (i.e., direct energy), idle periods (i.e., unload energy), and by auxiliary facilities (i.e., indirect energy).

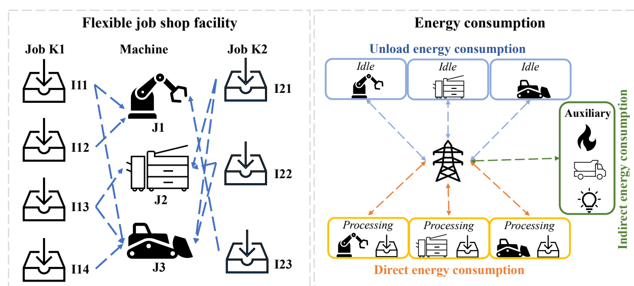

Figure 1. A FJSSP considering energy consumption.

To save energy, a machine j can be temporarily switched off or maintain standby after processing (see Figure 2). Its switch off/on energy and unload power are expressed by E_j^{O} and P_j^{U}, respectively.

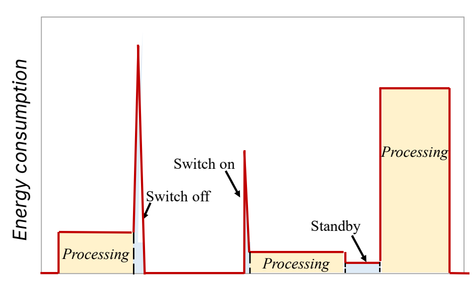

Figure 2. Energy consumption profile for a machine

MATHEMATICAL FORMULATION

In the global sequence-based formulation (**MG**), we define four-index binary variables to denote the global precedence of two operations (e.g. i and i'),

$$x_{kik'i'} = \begin{cases} 1 & \text{if } i \text{ in } k \text{ precedes } i' \text{ in } k' \text{ on one machine} \\ 0 & \text{otherwise} \end{cases}$$

Binary variables w_{kij} express the assignment of operations on machines,

$$w_{kij} = \begin{cases} 1 & \text{if } i \text{ in } k \text{ is assigned on a machine } j \\ 0 & \text{otherwise} \end{cases}$$

Any operation i of job k must be assigned on a machine j.

$$\sum_{j \in \mathbf{J}_{ki}} w_{kij} = 1 \qquad \forall k, i \in \mathbf{I}_k \tag{1}$$

If operations i, i' are processed on a machine, either i precedes i' or i' precedes i. That is if variables $w_{kij} = 1$ and $w_{k'i'j} = 1$, then exactly one of $x_{kik'i'}$ or $x_{k'i'ki}$ should be equal to 1, indicating their precedence on j.

$$x_{kik'i'} + x_{k'i'ki} \geq w_{kij} + w_{k'i'j} - 1 \quad \forall k, k', j \in (\mathbf{J}_{ki} \cup \mathbf{J}_{k'i'}), i \in \mathbf{I}_k, i' \in \mathbf{I}_{k'}, (k < k') \cup (k = k' \cap i < i') \tag{2}$$

When two operations i in job k and i' in job k' are performed on different machines (e.g., $w_{kij} = 1$ and $w_{k'i'j} = 0$), their corresponding variables $x_{k'i'ki}$ and $x_{kik'i'}$ should be zero. The set $i' \in \mathbf{I}^{\mathrm{G}}_{ki}$ includes operations i' that can share the same machine with i of k.

$$w_{k'i'j} \geq w_{kij} + x_{k'i'ki} + x_{kik'i'} - 1 \quad \forall k, k', j \in (\mathbf{J}_{ki} \cup \mathbf{J}_{k'i'}), i \in \mathbf{I}_k, i' \in \left(\mathbf{I}_{k'} \cap \mathbf{I}^{\mathrm{G}}_{ki}\right), [k < k' \cup (k = k' \cap i < i')] \tag{3}$$

$$w_{kij} \geq w_{k'i'j} + x_{k'i'ki} + x_{kik'i'} - 1 \quad \forall k, k', j \in (\mathbf{J}_{ki} \cup \mathbf{J}_{k'i'}), i \in \mathbf{I}_k, i' \in \left(\mathbf{I}_{k'} \cap \mathbf{I}^{\mathrm{G}}_{ki}\right), [k < k' \cup (k = k' \cap i < i')] \tag{4}$$

Continuous variable T_{ki} denotes the start time for operation i of job k. An operation $i \in \mathbf{I}_k$, which is the successor of the operation $i' \in \mathbf{I}_k$ in the same job, should start after the completion of operation i'.

$$T_{ki} \geq T_{ki'} + \sum_{j \in \mathbf{J}_{ki'}} \left(P^{\mathrm{T}}_{ki'j} \cdot w_{ki'j}\right) \quad \forall k, i' \in \mathbf{I}_k, i \in \mathbf{I}_k, i = i' + 1 \tag{5}$$

We define binary variables z_{kij} and 0-1 continuous variables y_{kij} to express the machine modes on j after processing an operation i of job k.

$$z_{kij} = \begin{cases} 1 & \text{if } j \text{ is switched off after processing } i \text{ in } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{kij} = \begin{cases} 1 & \text{if } j \text{ remains standby after processing } i \text{ in } k \\ 0 & \text{otherwise} \end{cases}$$

A machine j would remain standby or be switched off after performing i (see eq. 6). It ensures that y_{kij} is 0 or 1 only even it is defined as 0-1 continuous variables.

$$y_{kij} + z_{kij} = w_{kij} \qquad \forall k, i, j \in \mathbf{J}_{ki} \tag{6}$$

Continuous variables ST_{kij} denote the duration of a machine j keeping standby mode after processing $i \in \mathbf{I}_k$. ST_{kij} equals the time interval between the finish of operation i and the start of its immediate precedence operation $i' \in \mathbf{I}_{k'}$ on the identical machine j, as handled in Eq. (7) and Eq. (8). Specifically, Eq. (7) handles that if a machine remains standby after performing operation i of job k (i.e., $z_{kij} = 0$), the start time of any subsequent operation i' (i.e., $x_{kik'i'} = 1$) is expected to be later than the finish time of i plus the standby duration.

$$T_{k'i'} \geq T_{ki} + \sum_{j \in \mathbf{J}_{ki}} \left(P^{\mathrm{T}}_{kij} \cdot w_{kij} + ST_{kij} + \frac{E^{\mathrm{O}}_j}{P^{\mathrm{U}}_j} \cdot z_{kij}\right) - M \cdot (1 - x_{kik'i'}) \quad \forall k, k', (k \neq k') \cup (k = k' \cap i < i'), i \in \mathbf{I}_k, i' \in \left(\mathbf{I}_{k'} \cap \mathbf{I}^{\mathrm{G}}_{ki}\right) \tag{7}$$

The immediate successor i' of operation i on the same machine is identified by Eq. (8), where the number of preceding operations of i' should be one more that of operation i. Here, the M indicates a big number, taking the value of 10000 in this work. N^{max}_j is the maximum number of jobs that can be processed on j.

$$T_{k'i'} \leq T_{ki} + \sum_{j \in \mathbf{J}_{ki}} \left(P^{\mathrm{T}}_{kij} \cdot w_{kij} + ST_{kij} + \frac{E^{\mathrm{O}}_j}{P^{\mathrm{U}}_j} \cdot z_{kij}\right) + M \cdot \left[\max_j\left(N^{max}_j\right)\right] \cdot (1 - x_{kik'i'}) + M \cdot \left(1 - \sum_{j \in \mathbf{J}_{ki}} y_{kij}\right) + M \cdot \left(\sum_{k''} \sum_{\substack{i'' \in \mathbf{I}^{\mathrm{G}}_{k'i'} \\ (k'' \neq k') \cup (k'' = k' \cap i'' \neq i')}} x_{k''i''k'i'} - \sum_{k''} \sum_{\substack{i'' \in \mathbf{I}^{\mathrm{G}}_{ki} \\ (k'' \neq k) \cup (k'' = k \cap i'' \neq i)}} x_{k''i''ki} - 1\right) \quad \forall k, k', (k \neq k') \cup (k = k' \cap i < i'), i \in \mathbf{I}_k, i' \in \left(\mathbf{I}_{k'} \cap \mathbf{I}^{\mathrm{G}}_{ki}\right) \tag{8}$$

Standby energy consumption ES_{ki} is calculated using Eq. (9), which is proportional to the duration of standby time and the unload power of the machine j. ST_{kij} is enforced as zero if the machine does not remain standby after processing $i \in \mathbf{I}_k$, as indicated in Eq. (10).

$$ES_{ki} = \sum_{j \in \mathbf{J}_{ki}} \left(ST_{kij} \cdot P^{\mathrm{U}}_j\right) \qquad \forall k, i \in \mathbf{I}_k \tag{9}$$

$$ST_{kij} \leq \left[\min\left(M, \frac{E^{\mathrm{O}}_j}{P^{\mathrm{U}}_j}\right)\right] \cdot y_{kij} \qquad \forall k, i, j \in \mathbf{J}_{ki} \tag{10}$$

Makespan should be larger than the finish time of the last operation ($\mathbf{LI}_k$) in all jobs. And it is no less than cumulative processing times of all operations in any job.

$$T_{ki} + \sum_{j \in \mathbf{J}_{ki}} \left(w_{kij} \cdot P^{\mathrm{T}}_{kij}\right) \leq MS \qquad \forall k, i \in (\mathbf{I}_k \cap \mathbf{LI}_k) \tag{11}$$

$$\sum_{i \in \mathbf{I}_k} \sum_{j \in \mathbf{J}_{ki}} \left(w_{kij} \cdot P^{\mathrm{T}}_{kij}\right) \leq MS \qquad \forall k \tag{12}$$

On any j, the accumulated time used to process operations and keep standby should be no larger than MS.

$$\sum_k \sum_{i \in \mathbf{I}_{kj}} \left(w_{kij} \cdot P^{\mathrm{T}}_{kij} + ST_{kij} + \frac{E^{\mathrm{O}}_j}{P^{\mathrm{U}}_j} \cdot z_{kij}\right) \leq MS \quad \forall j \tag{13}$$

Constraint (14) is the objective function.

$$TEC = \sum_k \sum_i \sum_{j\in\mathbf{J}_{ki}} \left(w_{kij} \cdot P^{\mathrm{T}}_{kij} \cdot P^{\mathrm{C}}_{kij}\right) + \beta \cdot MS + \sum_k \sum_{i\in\mathbf{I}_k} ES_{ki} + \sum_i \sum_k \sum_{j\in\mathbf{J}_{ki}} \left(E^{\mathrm{O}}_j \cdot z_{kij}\right) \quad (14)$$

The start time of an operation i of job k should be always after the minimum required time for all predecessor operations in the same job.

$$T_{ki} \geq \sum_{i'<i,i'\in\mathbf{I}_k} \left\{\min_{j\in\mathbf{J}_{ki}}\left(P^{\mathrm{T}}_{ki'j}\right)\right\} \quad \forall k, i \in \mathbf{I}_k \quad (15)$$

Upper bounds are imposed on the positive variables MS and T_{ki}, as specified by Eq. (16) and Eq. (17).

$$MS \leq M \quad (16)$$

$$T_{ki} \leq M \quad \forall k, i \quad (17)$$

Moreover, if an operation cannot be performed on a machine j (i.e., $j \notin \mathbf{J}_{ki}$), its corresponding binary variables (e.g., w_{kij}, y_{kij} and z_{kij}) should be set to zero.

$$w_{kij} = 0, y_{kij} = 0, z_{kij} = 0 \quad \forall k, i, j \notin \mathbf{J}_{ki} \quad (18)$$

Precedence between operations of the same job can be confirmed on the same machine. And variables $x_{kik'i'}$ are fixed to zero if i' cannot share any machine with i (i.e., $i' \notin \mathbf{I}^{\mathrm{G}}_{ki}$)

$$x_{ki'ki} = 0 \quad \forall i \in \mathbf{I}_k, i' \in \mathbf{I}_k, i < i' \quad (19)$$

$$x_{kik'i'} = 0 \quad \forall k, i \in \mathbf{I}_k, k', i' \in \mathbf{I}_{k'}, i' \notin \mathbf{I}^{\mathrm{G}}_{ki} \quad (20)$$

Finally, equations (21-23) list all the continuous and binary variables in model **MG**. We complete the mathematical model **MG** consisting of constraints (1-23).

$$ES_{ki}, ST_{kij}, MS, T_{ki} \geq 0 \quad (21)$$

$$0 \leq y_{kij} \leq 1 \quad (22)$$

$$z_{kij}, x_{kik'i'}, w_{kij} \in \{0,1\} \quad (23)$$

COMPUTATIONAL STUDIES

Table 1: Problem size of benchmark examples.

Problem size ($K * I * J$)	Example (Ex)
2*4*2	1-5
2*6*3	6-10
3*6*2	11-15
3*9*3	16-20
6*36*6	21
10*100*10	22, 39-43
10*50*5	24-28
15*75*5	29-33
20*100*5	34-38
15*150*10	44-48
20*200*10	49-54
30*300*10	55-58

We utilize the proposed **MG** to solve 58 benchmark examples from the literature [20,21] to evaluate its computational performance. These examples are sourced from the Hurink_DATA dataset [20], comprising 20 small-scale and 38 industrial-scale examples. Their respective problem sizes are listed in Table 1. **MG** is benchmarked against MILP-based model formulated using local sequence-based representations (**ML**) [24] and event-based continuous-time representations (**MU**) [21]. Additionally, the solution quality of **MG** is compared with a decomposition approach from Rakovitis *et al.* [21]. All examples are solved using CPLEX22.1.0/GAMS40.3.0 on a desktop computer with AMD RyzenTM 9 5950X 3.40 GHz and 96.0 GB RAM running Windows 10. A time limit of one hour is imposed for all MILP formulations.

Comparison with mathematical programming models

Comparisons between **MG** and MILP model from literature for the small-size examples are provided in Table 2. All formulations generate identical optimal solutions, which represent the global optimum, with similar computational times of approximately 1 second.

Table 2: TEC computational results for small-size examples from **MG**, **MU** and **ML**.

Ex	MG, MU, ML	Ex	MG, MU, ML
1	63.03	*11*	166.23
2	122.44	*12*	176.75
3	75.74	*13*	121.30
4	146.63	*14*	156.86
5	78.40	*15*	163.20
6	220.74	*16*	219.46
7	97.54	*17*	306.68
8	146.81	*18*	210.60
9	230.66	*19*	269.52
10	161.06	*20*	274.94

Table 3: TEC computational time for industrial-size examples from **MG**, **MU** and **ML**

Ex	MG	MU	ML	Ex	MG	MU	ML
21	**183**	183	183	*40*	**3327**	3705	3715
22	3473	3779	3393	*41*	**3344**	3618	3497
23	3639	3421	3862	*42*	**3698**	3731	3798
24	**1776**	1776	1776	*43*	**3692**	3659	4173
25	**1769**	1830	1800	*44*	**5052**	5091	NA
26	**1530**	1623	1603	*45*	4722	4612	NA
27	**1647**	1647	1691	*46*	**5197**	5225	NA
28	**1463**	1465	1495	*47*	**5314**	5467	NA
29	**2584**	2584	2626	*48*	**5362**	5291	NA
30	2404	2439	2378	*49*	**6815**	7582	NA
31	**2486**	2486	2568	*50*	**7772**	8823	NA
32	**2638**	2638	2696	*51*	**7059**	8826	NA
33	**2522**	2522	2522	*52*	**7854**	NA	NA
34	3744	3353	3973	*53*	**7075**	NA	NA
35	**2996**	3036	3277	*54*	**14480**	NA	NA
36	**3162**	3195	3309	*55*	**13930**	NA	NA
37	3473	3422	3595	*56*	**19081**	NA	NA
38	3457	3441	NA	*57*	**15674**	NA	NA
39	3781	3800	3713	*58*	**21380**	NA	NA

Table 3 presents the optimized TEC of industrial-size examples from MILP formulations. If **MG** achieves solutions that are no worse than those from **MU** and **ML**, its TEC results are highlighted in bold. And 'NA' indicates no available solutions are found. Table 3 proves that **MG** outperforms both **MU** and **ML** in examples 25, 26, 28, 35, 36, 40-44, 46, 47, and 49-58. The maximum reduction in TEC is 20% (7059 vs. 8826) in example 51, compared to **MU**. Advantage of **MG** is mainly evident in complicated examples. In detail, for cases with more than 200 operations (e.g., examples 49-58), both **MU** and **ML** fail to find feasible solutions in most instances. However, **MG** successfully provides feasible solutions for these cases.

Some examples converge within a computation time of less than 1 hour. The specific computational times for these examples are presented in Table 4, allowing for a comparison of the computational efficiency of the MILP formulations. Table 4 demonstrates that **MG** achieves shorter computational time relative to **MU** and **ML** in most cases (8 out of 9 examples). The most significant reduction is observed in example 26, where **MG** requires 1/14 of the time (248 seconds vs. 3600 seconds) to reach a better solution (1530 vs. 1623 and 1603).

Table 4: Computational times (Seconds) of industrial-size examples from **MG**, **MU** and **ML**.

Ex	MG	MU	ML	Ex	MG	MU	ML
24	**5**	97	325	*32*	**362**	3600	3600
26	**248**	3600	3600	*33*	**218**	1668	3326
28	**209**	3600	3600	*35*	**3336**	3600	3600
29	644	206	3600	*36*	**1481**	3600	3600
31	**526**	1136	3600				

Comparison with a decomposition algorithm

A decomposition algorithm [21], denoted by **GD**, is proposed to address industrial-scale problems. We compare its performance with the proposed **MG** (see Table 5). Since the computational time for **MG** is currently limited to one hour, making it suitable for real-world applications in complex cases, our focus is primarily on comparing the solution quality between **GD** and **MG**. Table 5 demonstrates the clear superiority of **MG** for examples with no more than 100 operations (i.e., examples 21-43), achieving a maximum TEC reduction of 13.6% (2996 vs. 3467 in example 35). For examples with more than 300 operations, although **MG** is capable of finding feasible solutions, it yields worse optimal solutions relative to **GD**.

CONCLUSIONS

In this work, we presented a mixed-integer linear programming formulation for energy-efficient flexible job shop scheduling problems. The objective was to minimize total energy consumption, including energy used during processing operations, idle durations, and auxiliary facilities. The formulation was developed based on the global precedence relationships of operations on machines. Since immediate precedence relationships cannot be directly expressed using decision variables, we incorporated novel constraints to identify them. This enabled the precise estimation of idle durations between adjacent operations on machines, which in turn determined the optimal machine modes. Computational studies showed the superiority of the proposed formulation. Its advantages are reflected in reduction in total energy consumption by up to 20%. It successfully provided feasible solutions for all considered examples, including those where existing mathematical formulations failed.

Table 5: TEC computational results of industrial-size examples from **MG** and **GD**.

Ex	MG	GD	Ex	MG	GD
21	**183**	196	*40*	**3327**	3564
22	**3473**	3850	*41*	**3344**	3517
23	**3639**	4063	*42*	**3698**	3682
24	**1776**	1837	*43*	**3692**	3959
25	**1769**	1922	*44*	**5052**	5335
26	**1530**	1682	*45*	**4722**	4978
27	**1647**	1764	*46*	5197	5180
28	**1463**	1570	*47*	5314	4993
29	**2584**	2694	*48*	**5362**	5567
30	**2404**	2541	*49*	**6815**	7164
31	**2486**	2682	*50*	7772	7758
32	**2638**	2728	*51*	7059	6683
33	**2522**	2750	*52*	7854	7241
34	**3744**	3589	*53*	**7075**	7229
35	**2996**	3467	*54*	14480	8712
36	**3162**	3397	*55*	13930	9690
37	**3473**	3767	*56*	19081	9091
38	**3457**	3864	*57*	15674	8873
39	**3781**	3887	*58*	21380	10004

ACKNOWLEDGEMENTS

Authors appreciate financial support from China Scholarship Council and The University of Manchester (201908130170, 202106440020) and Engineering and Physical Sciences Research Council (EP/T03145X/1, EP/V051008/1).

REFERENCES

1. Yang L, Li J, Chao F, Hackney P, Flanagan M. Job shop planning and scheduling for manufacturers with manual operations. Expert Syst 38: e12315 (2021) https://doi.org/10.1111/exsy.12315
2. Liao J, Lin C. Optimization and simulation of job-shop supply chain scheduling in manufacturing enterprises based on particle swarm optimization. Int J Simul Model 18:187-196 (2019)

https://doi.org/10.2507/IJSIMM18(1)C05
3. Nouri HE, Driss OB, Ghédira K. A classification schema for the job shop scheduling problem with transportation resources: state-of-the-art review. Artif Intell Perspect Intell Syst Adv Intell Syst Comput 464:1-11 (2016) https://doi.org/10.1007/978-3-319-33625-1_1
4. Tian Z, Jiang X, Liu W, Li Z. Dynamic energy-efficient scheduling of multi-variety and small batch flexible job-shop: a case study for the aerospace industry. Comput Ind Eng 178:109111 (2023) https://doi.org/10.1016/j.cie.2023.109111
5. Jiang X, Tian Z, Liu W, Suo Y, Chen K, Xu X, Li Z. Energy-efficient scheduling of flexible job shops with complex processes: a case study for the aerospace industry complex components in China. J Ind Inf Integr 27:100293 (2022) https://doi.org/10.1016/j.jii.2021.100293
6. Brucker P, Schlie R. Job-shop scheduling with multipurpose machines. Computing 45:369-375 (1990) https://doi.org/10.1007/BF02238804
7. Demir Y, İşleyen SK. Evaluation of mathematical models for flexible job-shop scheduling problems. Appl Math Model 37:977-988 (2013) https://doi.org/10.1016/j.apm.2012.03.020
8. Ham A, Park MJ, Kim KM. Energy-aware flexible job shop scheduling using mixed integer programming and constraint programming. Math Probl Eng 2021:1-12 https://doi.org/10.1155/2021/8035806
9. Wang S, Yu J. An effective heuristic for flexible job-shop scheduling problem with maintenance activities. Comput Ind Eng 59:436-447 (2010) https://doi.org/10.1016/j.cie.2010.05.016
10. Li X, Gao L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. Int J Prod Econ 174:93-110 (2016) https://doi.org/10.1016/j.ijpe.2016.01.016
11. Ding H, Gu X. Improved particle swarm optimization algorithm based novel encoding and decoding schemes for flexible job shop scheduling problem. Comput Oper Res 121:104951 (2020) https://doi.org/10.1016/j.cor.2020.104951
12. Xie J, Gao L, Peng K, Li X, Li H. Review on flexible job shop scheduling. IET Collab Intell Manuf 1:67-77 (2019) https://doi.org/10.1049/iet-cim.2018.0009
13. Coelho P, Pinto A, Moniz S, Silva C. Thirty years of flexible job-shop scheduling: a bibliometric study. Procedia Comput Sci 180:787-796 (2021) https://doi.org/10.1016/j.procs.2021.01.329
14. U.S. Energy Information Administration. https://www.eia.gov/energyexplained/use-of-energy/industry.php (2023) [13 July 2023].
15. Nguyen TAL, Tran LNH, Paris H, Mathieu M. Evaluating energy efficiency of production machine. Appl Mech Mater 889:580-587 (2019) http://doi.org/10.4028/www.scientific.net/AMM.889.580
16. Meng L, Ren Y, Zhang B, Li JQ, Sang H, Zhang C. MILP modeling and optimization of energy-efficient distributed flexible job shop scheduling problem. IEEE Access 8:3032548 (2020) https://doi.org/10.1109/ACCESS.2020.3032548
17. Meng L, Zhang C, Shao X, Ren Y. MILP models for energy-aware flexible job shop scheduling problem. J Clean Prod 210:118289 (2019) https://doi.org/10.1016/j.jclepro.2018.11.021
18. Zhang L, Li Z, Królczyk G, Wu D, Tang Q. Mathematical modeling and multi-attribute rule mining for energy efficient job-shop scheduling. J Clean Prod 241:118289 (2019) https://doi.org/10.1016/j.jclepro.2019.118289
19. Park MJ, Ham A. Energy-aware flexible job shop scheduling under time-of-use pricing. Int J Prod Econ 248:108507 (2022) https://doi.org/10.1016/j.ijpe.2022.108507
20. Zhang L, Tang Q, Wu Z, Wang F. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. Energy 138:101421 (2017) https://doi.org/10.1016/j.energy.2017.07.005
21. Rakovitis N, Li D, Zhang N, Li J, Zhang L, Xiao X. Novel approach to energy-efficient flexible job-shop scheduling problems. Energy 238:121773 (2022)https://doi.org/10.1016/j.energy.2021.121773
22. Dai M, Tang D, Giret A, Salido MA. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robot Comput-Integr Manuf 59:101249 (2019) https://doi.org/10.1016/j.rcim.2019.04.006
23. Cheng L, Tang Q, Zhang L, Meng K. Mathematical model and enhanced cooperative co-evolutionary algorithm for scheduling energy-efficient manufacturing cell. J Clean Prod 326:129248 (2021)https://doi.org/10.1016/j.jclepro.2021.129248
24. Li D, Zheng T, Li J, Teymourifar A. A hybrid framework integrating machine-learning and mathematical programming approaches for sustainable scheduling of flexible job-shop problems. Chem Eng Trans 103:385-390 (2023) https://doi.org/10.3303/CET23103065
25. Gong G, Tang J, Huang D, Luo Q, Zhu K, Peng N. Energy-efficient flexible job shop scheduling problem considering discrete operation sequence flexibility. Swarm Evol Comput 84:101421 (2024) https://doi.org/10.1016/j.swevo.2023.101421