CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Energy-aware blocking hybrid flow shop scheduling problem with sequence-depend setup times and machine speed levels
Authors	Missaoui, Ahmed;Ozturk, Cemalettin;O'Sullivan, Barry
Publication date	2025
Original Citation	Missaoui, A., Ozturk, C. and O'Sullivan, B. (2025) 'Energy-aware blocking hybrid flow shop scheduling problem with sequence-depend setup times and machine speed levels', Procedia Computer Science, 253, pp. 1134–1143. https://doi.org/10.1016/j.procs.2025.01.175
Type of publication	Article (peer-reviewed)
Link to publisher's version	10.1016/j.procs.2025.01.175
Rights	© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0) Peer-review under responsibility of the scientific committee of the 6th International Conference on Industry 4.0 and Smart Manufacturing - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-09-21 20:12:31
Item downloaded from	https://hdl.handle.net/10468/17293

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 253 (2025) 1134–1143

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Industry 4.0 and Smart Manufacturing

Energy-Aware Blocking Hybrid Flow Shop Scheduling Problem with Sequence-depend Setup Times and Machine Speed Levels

Ahmed Missaoui[a], Cemalettin Ozturk[b], Barry O'Sullivan[c]

[a] *Insight Centre for Data Analytics, University College Cork, Ireland (e-mail: ahmed.missaoui@insight-centre.org).*
[b] *Process, Energy and Transport Engineering, Munster Technological University, Cork, Ireland (e-mail: cemalettin.ozturk@mtu.ie)*
[c] *Insight Centre for Data Analytics, University College Cork, Ireland (e-mail: b.osullivan@cs.ucc.ie).*

Abstract

The industry sector has the second largest energy demand after electricity generation. Given the high trend of energy prices, scarcity of its supply due to political instability and dependency on fossil fuels for its production makes energy the primary challenge for the manufacturing sector to stay competitive. Therefore, in addition to novel equipment technologies that use less energy, energy-efficient scheduling has also been a priority for manufacturing companies.In this study, the Hybrid Flowshop Scheduling Problem with Blocking Constraints and Sequence-Depend Setup Times (BHFS-SDST) is investigated for the minimization of makespan and total energy consumption (TEC). First, a novel bi-objective Mixed Integer Linear Programming (MILP) model is formulated and solved through augmented epsilon constraints. Then, a novel multi-objective approach based on Iterated Greedy meta-heuristic is developed for larger instances. Efficiency and the scalability of the proposed approaches are tested with small, medium, and large instances. Computational experiments show the effectiveness of developed methods in solving the BHFS-SDST problem.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 6th International Conference on Industry 4.0 and Smart Manufacturing

Keywords: Hybrid flowshop; blocking constraint; Sequence-depend setup times; Makespan; Energy consumption; Mixed Integer Linear Programming; Iterated greedy;

1. Introduction

Under the energy crises and climate change, manufacturing companies are exploring reducing energy consumption and carbon footprints. To achieve these objectives, companies have the choice of investing in extra equipment or improving the management of current equipment which is generally the cheapest option. Within the scope of the

* Ahmed Missaoui
E-mail address: ahmed.missaoui@insight-centre.org

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 6th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2025.01.175

second option, energy-efficient scheduling stands out as a crucial strategy for industry, reflecting in the increasing number of research publications on energy-efficient scheduling. A recent review study on energy-efficient manufacturing scheduling in the last decade [12] highlights the diversity of applications from the metal [5] to the food [22] industries and indicate the significance of energy concerns.

Most discrete manufacturing processes are organized into a series of stages, where the job is processed sequentially in each stage, named flow shop systems. Flow shops can experience reduced productivity due to bottlenecks at slower downstream stages [15]. Introducing parallel machines at each stage is one way to resolve this problem, which converts the flow shop into a hybrid flow shop [17]. Formally, a hybrid flow shop scheduling (HFS) problem consists of a set of n jobs that have to be processed in $g > 1$ stages. Each stage includes m_g machines where $m_g > 1$ in one stage at least. Many industries suffer from a lack of sufficient buffer areas between stages. This can cause machines to be blocked from releasing completed jobs until the availability of one machine in the next stages, which is studied in the literature as blocking constraints [10], This can cause unproductive energy consumption that needs to be minimized. Another significant unproductive energy consumption during the manufacturing process occurs during the changeover times between jobs in each machine due to cleaning and machine setup that also known as sequence-dependent setup times (SDST) [13]. Therefore, different energy consumption rates, including those for processing, idling, blocking, and setup, should be minimized. Simultaneously, industries need to enhance productivity by reducing the total completion time, referred to as makespan in the literature. In summary, the problem under investigation in this paper is the energy-aware multi-objective HFS with SDST and blocking constraints, denoted as BHFS-SDST with speed levels where the objective is simultaneously minimization of the makespan and total energy consumption. Balancing the two conflicting objectives is adding additional computational challenges to the problem under study which require novel modeling and solution methods. The remainder of this paper is organized as follows: Section 2 reviews the recent literature. The MILP model and an illustrative example are presented in Section 3. Subsequently, the proposed multi-objective Iterated Greedy algorithm is described in Section 4. Section 5 presents numerical experiments and discusses results. Finally, the conclusion and future research challenges are outlined in Section 6.

2. Related Work

Energy-aware scheduling has recently received considerable attention and different problems have been addressed using both single and multi-objective approaches. The HFS problem has been introduced with different constraints and energy aspects more than 58 times in the last decade according to the survey of [12]. Aiming to minimize the makespan under energy constraints, the HFS was investigated for the first time in [1]. [6] introduced a Lexicographic approach for energy-efficient Open shop with SDST and operation speed levels. [8] addressed the HFS with makespan and total energy cost using a multi-objective ant colony optimization method. In [23], the HFS is tackled with a multi-objective approach based on decomposition aiming to minimize TEC and Makespan. A case study of HFS in a real-world glass-ceramics production has been tackled in [20] for the minimization of Makespan and TEC, and a MILP model has been developed for the resolution. [18] proposed a multi-objective iterated local search to solve the HFS to minimize the makespan, total energy costs, and peak load. [19] investigated the energy-aware HFS and proposed a hybrid Archimedes optimization algorithm for the minimization of TEC. [21] addressed the HFS with SDST aiming to minimize both the Makespan and TEC using an improved memetic algorithm and a MILP model. [7] proposed a collaboration-based multi-objective algorithm for the minimization of makespan and TEC in a distributed HFS with machine velocity and resource constraints. Recently, [11] addressed the HFS with blocking constraints, aiming to minimize the makespan and TEC using the NSGA-II algorithm from [3] and the IG method from [4]. These two classic metaheuristics demonstrated comparable performance but were not compared with an exact method. To the best of our knowledge, this is the first study that considers the energy-aware multi-objective HFS with SDST and blocking constraints (BHFS-SDST) with speed levels and distinct energy consumption rates.

3. Problem Statement

3.1. Mixed Integer Linear Programming

In the following, we introduce a novel MILP model for the HFS problem with blocking constraints and sequence-depend setup times extended from [16].

Parameters:
N : Set of real (non-dummy) jobs indexed by j, k, h.
N_0: $N \cup$ 0 set of all jobs including dummy first job in each machine
n : Number of jobs indexed by j, k, h.
g: Number of stages.
M : Set of stages indexed by i
vt : Set of speed indexed by v.
$speed_v$: speed v.
$U_v = speed_v \times speed_v$ The energy consumption per unit of time when machine m runs at speed v
m_i : Number of identical machines on stage i indexed by l.
p_{ij} : Standard processing time of job j at stage i.
st_{jki} : Setup times between job j and job k at stage i
Eid_i: Idle energy consumption per unit time at stage i.
Eb_i: Blocking energy consumption per unit time at stage i.
Est_i : Setup energy consumption per unit time at stage i.
H: Large integer.

Decision variables:
Let $\mathcal{R} = \mathbb{R}^+ \cup \{0\}$
$Cmax$: Makespan ,$\in \mathcal{R}$
TEC : Total energy consumption ,$\in \mathcal{R}$
S_{ij} :Starting time of job j at stage i,$\in \mathcal{R}$
C_{ij} :Completion time of job j at stage i,$\in \mathcal{R}$
D_{ij} :Departure time of job j at stage i,$\in \mathcal{R}$
B_{ij} :Blocking time of job j at stage i,$\in \mathcal{R}$
$ES_{i,l}$: Turn-on time of machine l at stage i,$\in \mathcal{R}$
$LC_{i,l}$: Turn-off time of machine l at stage i,$\in \mathcal{R}$
$Mst_{i,l}$: Setup times of machine l at stage i,$\in \mathcal{R}$
Mst_i : Sum of setup times at stage i,$\in \mathcal{R}$
Idl_i : Sum of idle time at stage i,$\in \mathcal{R}$
B_i : sum of blocking time at stage i,$\in \mathcal{R}$
Rp_{ij} : Actual processing time of job j at stage i,$\in \mathcal{R}$
EC_p : Processing energy consumption,$\in \mathcal{R}$
EC_{St} Setup energy consumption,$\in \mathcal{R}$
EC_B : Blocking energy consumption,$\in \mathcal{R}$
EC_{Id} Idle energy consumption,$\in \mathcal{R}$
Rp_i: Sum of actual processing time at stage i,$\in \mathcal{R}$
$X_{iljk} = 1$, if job j is the immediate predecessor of job k on machine l at stage i and 0, otherwise.

$V_{jkv} = 1$, if job j is processed at stage k with speed v and 0 Otherwise.

Mathematical model:

$$Min\ Cmax \tag{1}$$

$$Min\ TEC \tag{2}$$

$$\sum_{j \in N_0, j \neq k} \sum_{l \in m_i} X_{iljk} = 1, \forall i \in M, \forall k \in N \tag{3}$$

$$\sum_{j \in N, j \neq k} \sum_{l \in m_i} X_{ilkj} \leq 1, \forall i \in M, \forall k \in N, \tag{4}$$

$$\sum_{h\in N_0, h\neq k, h\neq j} X_{ilhj} \geq X_{iljk}, \forall i \in M, \forall l \in m_i, \forall j,k \in N, j \neq k, \tag{5}$$

$$\sum_{l\in m_i} (X_{iljk} + X_{ilkj}) \leq 1, \forall i \in M, \forall j \in N, k = j+1, \ldots, n, \tag{6}$$

$$\sum_{k\in N} X_{il0k} \leq 1, \forall i \in M, \forall l \in m_i \tag{7}$$

$$\sum_{v\in vt} V_{jiv} = 1, I\forall \in M, \forall j \in N \tag{8}$$

$$Rp_{ij} = \sum_{v\in vt} (V_{jiv} \times p_{ij} \div speed_v), \forall i \in M, \forall j \in N \tag{9}$$

$$C_{i0} = 0, \forall i \in M, \tag{10}$$

$$C_{1k} + H \times (1 - X_{1ljk}) \geq D_{1j} + Rp_{1k} + St_{jk1}, \forall l \in m_1, \forall j \in N_0, \forall k \in N, \forall k \neq j \tag{11}$$

$$C_{ik} + H \times (1 - X_{iljk}) \geq D_{ij} + St_{jki} + Rp_{ik} \forall i = 2, \ldots, m, \forall l \in m_i, \forall j \in N_0, \forall k \in N, \forall j \neq k \tag{12}$$

$$C_{ij} = S_{ij} + Rp_{ij}, i \in M, j \in N_0 \tag{13}$$

$$D_{ij} = C_{ij} + B_{ij}, \forall i \in M, \forall j \in N_0 \tag{14}$$

$$D_{ij} = S_{i+1,j}, \forall i \in M-1, \forall j \in N_0 \tag{15}$$

$$S_{i,j} - St_{0ji} \times X_{i,l,0,j} \leq ES_{k,l} + H \times (1 - X_{i,l,0,j}), \forall j \in N, \forall l \in \{m_k\}, \forall i \in M \tag{16}$$

$$S_{i,j} - St_{0ji}.X_{i,l,0,j} \geq ES_{k,l} - H \times (1 - X_{i,l,0,j}), \forall j \in N, \forall l \in \{m_k\}, \forall i \in M \tag{17}$$

$$C_{i,j} \leq LC_{i,l} + H \times (1 - X_{i,l,j,k}), \forall j,k \in N, \forall l \in \{m_k\}, \forall i \in M \tag{18}$$

$$Mst_{i,l} = \sum_{j\in N_0, j\neq k} \sum_{k\in N} (st_{jki} \times X_{i,l,j,k}), \forall l \in \{m_k\}, \forall i \in M \tag{19}$$

$$EC_p = \sum_{i\in M} \sum_{j\in N} \sum_{v\in vt} (p_{ij} \div speed_v) \times V_{jiv} \times U_v \times 4) \tag{20}$$

$$Mst_i = \sum_{i\in M} \sum_{l\in m_i} (Mst_{i,l}) \tag{21}$$

$$B_i = \sum_{i\in M} B_{i,j}, \forall j \in N \tag{22}$$

$$Rp_i = \sum_{i\in M} Rp_{i,j}, \forall j \in N \tag{23}$$

$$Idl_i = \sum_{i\in M} (LC_{i,l} - ES_{i,l}) - Rp_i - Mst_i - B_i \tag{24}$$

$$TEC = EC_p + \sum_{i\in M} (Mst_i \times Est_i + B_i \times Eb_i + Idl_i \times Eid_i) \tag{25}$$

$$S_{ij} \geq 0, C_{ij} \geq 0, D_{ij} \geq 0, \ i \in M, j \in N \tag{26}$$

$$X_{iljk} \in \{0,1\}, i \in M, l \in m_i, j \in N_0, k \in N, j \neq k \tag{27}$$

Table 1. Processing times (min) and energy consumption per stage (kWh

Consumption	Job	Stage 1	Stage 2
Processing	1	3	12
	2	4	10
	3	2	3
	4	6	6
	5	12	2
	6	9	2
Energy	Proces	4	2
	Idle	3	1
	Blocking	2	3
	Setup	3	5

Table 2. Speed levels

Job	1	2	3	4	5	6
$Stage1$	1	2	1	2	3	2
$Stage2$	3	2	1	3	1	1

Objective functions (1) and (2) refer to makespan and total energy consumption. Constraints (3) ensures that every job has to be preceded by exactly one job on only one machine at each stage. Only the relevant variables are considered in this context. It is important to mention that for every machine in every stage, a dummy job (indexed as 0) is introduced to precede the first job. Constraint set (4) ensures that every job has at most one successor. Constraint set (5) ensures that if a job is processed on a given machine at a stage, then it must have a predecessor on that same machine. Constraint set (6) forbids circular cross-precedence between non-dummy jobs In constraint set (7), we ensure that dummy job 0 can only be a predecessor of at most one job on each machine at each stage. Constraints (8) ensures that each job is exclusively processed at a single, predetermined speed level within a particular stage. Constraints (9) calculates the actual processing time of each job in all stages according to the selected speed level. The completion time is set to 0 for all these dummy jobs in all stages with constraint set (10). Constraint set (11) is used to calculate the completion times of jobs at the first stage. Similarly, constraint (12) performs the same function for the other stages. Constraints (13) through (15) describe the relationship between the starting time, completion time, and departure time of jobs. Constraints (16) and (17) specify the turn-on time of each machine in all stages, while constraint (18) identifies the turn-off time of machines in all stages. Constraints (19) calculates the setup times on each machine, and constraints (20) computes the processing energy consumption used in the work of [2]. Additionally, constraints (21) calculates the setup times per stage. Furthermore, constraints (22) calculates the blocking times per stage. Moreover, constraints (23) computes the actual processing times per stage. Constraints (24) calculates the idle time per stage. Constraints (25) is utilized to determine the total energy consumption. Finally, constraints (26) and (27) define the domain of the decision variables.

There are various solution approaches for addressing multi-objective mathematical models, including lexicographic, weighted sum, goal programming and the ϵ-constraint method. In this study, we employed the augmented ϵ-constraint method to exclusively derive Pareto-optimal solutions [9]). Similar to the standard ϵ-constraint approach, the augmented ϵ-constraint method optimizes only one of the objective functions, treating the others as constraints. However, in contrast to the classical ϵ-constraint method, the augmented one defines the objective function constraints as equalities. This is accomplished by introducing necessary slack/surplus variables. Subsequently, these slack/surplus variables are incorporated into the objective function as a second term to ensure Pareto optimality.

3.2. Illustrative Example

We provide an illustrative example to describe how to obtain the Makespan and the Total Energy Consumption (TEC) values for a small instance of six jobs and two stages, where each stage consists of two parallel machines. All necessary data is available in Tables 1, 2, and 3. Table 1 contains the processing times of the jobs and the energy

Table 3. Sequence dependent setup times (min) for the example.

Stage 1							Stage 2						
Jobs	1	2	3	4	5	6	Jobs	1	2	3	4	5	6
0	1	2	1	3	2	4	0	3	2	1	4	3	5
1	–	1	1	3	1	1	1	–	1	1	2	3	1
2	1	–	2	2	3	2	2	1	–	2	2	3	1
3	2	5	–	2	1	3	3	2	3	–	2	1	3
4	3	1	5	–	3	1	4	3	1	1	–	2	1
5	2	1	1	2	–	5	5	2	1	1	2	–	4
6	2	1	3	2	3	–	6	2	1	1	2	6	–

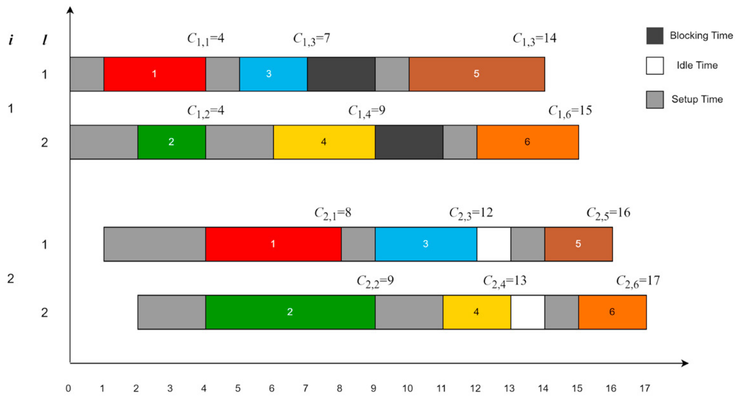

Fig. 1. Gantt chart of Seq (1, 2, 3, 4, 5, 6). Note the y-axis is indexed by i and l.

consumption for each machine state. The sequence-dependent setup times between jobs, including dummy jobs (job zero), can be found in Table 3 while Table 2 presents the energy speed levels for each job at each stage.

The resulting Gantt chart for the job sequence $\pi = (1, 2, 3, 4, 5, 6)$ is illustrated in Fig. 1, showing a Makespan of 17 min and a TEC of 538 kwh.

4. Refined Iterated Pareto Greedy

The multi-objective iterated greedy (MOIG) was introduced and applied for the first time in [4]. Subsequently, various versions of MOIG were proposed, demonstrating its effectiveness in addressing scheduling problems, which represents a motivation to choose this method. The proposed algorithm is referred to as Refined Iterated Pareto Greedy or (RIPG) and the main phases are given in Figure 2. The remainder of this section is dedicated to describing all steps in detail.

4.1. Initialization

As with all IG algorithms, the quality of the initial solution has an impact on its performance. In this paper, we started with one solution for each objective. The NEH heuristic of [14] is used for the Makespan objective while a modified version for the same heuristic is used in [12] for the TEC objective. In total, we have 2 initial solutions. Both of them will move to the Greedy phases to constitute the initial Pareto front.

4.2. Selection Phase

In this iterative process, each phase can get a set of non-dominated solutions, and the idea of processing all of them in the next phases looks expensive in terms of time and memory. To solve the problem, only one solution will

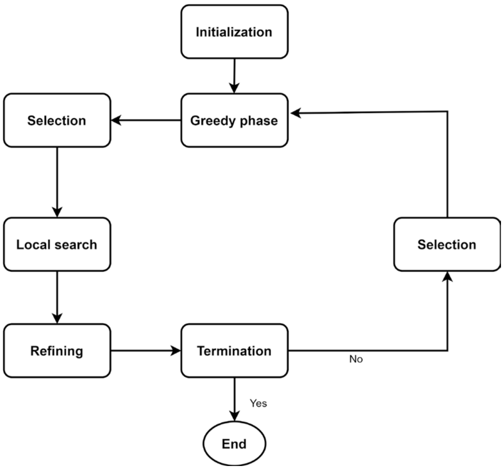

Fig. 2. Flowchart of RIPG

be selected to be handled by the next operator. The idea is to calculate the Crowding distance (CD) of each solution as introduced in [3]. The solution with maximum CD will be handled in the next operator.

4.3. Greedy Phase

Adopting the same process as outlined in [4], the approach involves initiating the destruction phase when a set of d random jobs needs to be removed from the selected solution and placed in a new set called D after a random perturbation of their speed levels. In the construction phase, these jobs are reintroduced one by one into all possible positions of the initial solution, forming a set of partial sequences. Inserting the next job into all partial solutions may not seem promising in terms of complexity, that's why only non-dominated solutions are retained for the subsequent insertion.

4.4. Local Search

Local search operators can improve the quality of solutions through intensification. In this phase, one job from the selected solution should be removed and reinserted in all possible positions after a random modification in its speed levels. As a result, we have a new set of n new sequences. Only the non-dominated solutions are kept and handled in the next step.

4.5. Refining Phase

At the end of each iteration, the resulting Pareto front undergoes processing during a refining phase based on Variable Neighborhood Descent (VND). The primary approach involves individually treating each solution using VND, incorporating insertion and interchange operations for L_S iterations knowing that the loop size L_S is a parameter. Finally, the dominant solution between the current and new ones is retained.

5. Computational Evaluation and Statistical Experimentation

5.1. Experimental Settings

To test the performance of the proposed methods, we generated three benchmarks of varying sizes, resulting in a total of 54 instances. These instances represent the combinations of three parameters: n is equal to $(6, 8, 10, 12)$, $(15, 20, 30)$ and $(50, 100)$ for respectively small, medium and large set of instances while $g = (2, 3, 4)$ and $m_i(2, 3)$. The processing time (min) is randomly generated $U \sim [1..99]$ while the energy consumption values kwh at each stage

Table 4. I_h and GD values for small instances (Best values are in bold)

			I_h		GD	
n	g	m	RIPG	Aug	RIPG	Aug
6	2	2	1.09	**1.28**	79.84	**0.00**
		3	0.88	**1.20**	146.71	**0.00**
	3	2	1.01	**1.27**	132.29	**0.00**
		3	0.92	**1.20**	109.80	**0.00**
	4	2	0.90	**1.29**	318.00	**0.00**
		3	0.87	**1.20**	132.36	**0.00**
			0.94	**1.24**	153.17	**0.00**
8	2	2	0.86	**1.16**	110.34	**0.00**
		3	0.90	**1.18**	119.57	**0.00**
	3	2	0.93	**1.20**	154.15	**0.00**
		3	0.94	**1.22**	124.96	**0.00**
	4	2	0.89	**1.23**	321.26	**34.96**
		3	1.12	**1.30**	355.13	**22.24**
			0.94	**1.21**	197.57	**9.53**
10	2	2	0.94	**1.23**	153.11	**0.00**
		3	1.12	**1.32**	180.12	**6.65**
	3	2	1.01	**1.24**	167.57	**17.23**
		3	0.80	**1.22**	400.24	**17.19**
	4	2	0.99	**1.28**	305.39	**111.47**
		3	0.88	**1.33**	529.99	**14.19**
			0.96	**1.27**	289.40	**27.79**
12	2	2	0.95	**1.17**	178.99	**52.13**
		3	1.04	**1.20**	147.86	**8.52**
	3	2	1.02	**1.18**	179.95	**38.78**
		3	1.07	**1.25**	288.32	**52.53**
	4	2	0.88	**1.15**	619.71	**33.19**
		3	0.93	**1.15**	408.10	**124.54**
			0.98	**1.18**	303.82	**51.61**
Avg			0.96	**1.23**	235.99	**22.23**

Table 5. I_h and GD values for medium instances (Best values are in bold)

			I_h		GD	
n	g	m	RIPG	Aug	RIPG	Aug
15	2	2	0.92	**1.20**	155.41	**25.50**
		3	0.89	**1.02**	191.22	**111.37**
	3	2	0.97	**1.05**	113.64	**67.17**
		3	0.99	**1.01**	**91.03**	120.69
	4	2	0.92	**1.13**	655.10	**434.00**
		3	0.91	**1.03**	240.69	**73.31**
			0.93	**1.07**	241.18	**138.67**
20	2	2	**1.04**	0.96	**58.53**	183.53
		3	**1.07**	0.96	**130.49**	357.45
	3	2	1.06	**1.11**	113.35	**115.41**
		3	**1.02**	0.99	**73.57**	173.88
	4	2	1.04	**1.05**	128.85	**106.67**
			1.04	1.01	**100.96**	187.39
30	2	2	**1.20**	1.05	**2.43**	562.49
		3	**1.20**	1.08	**1.29**	741.54
	3	2	**1.07**	0.98	**21.14**	457.84
		3	**1.15**	1.12	**19.96**	182.42
	4	2	1.03	**1.06**	**122.94**	233.19
		3	1.07	**1.12**	**73.97**	219.93
			1.11	1.08	**45.10**	373.91
Avg			1.03	**1.06**	**125.98**	243.68

are generated from uniform distributions of $U \sim [1..3]$, $U \sim [5..7]$, and $U \sim [3..5]$ for processing energy, blocking energy and idle energy respectively. The speed levels are given as $vt = (1.00, 1.30, 1.55, 1.75, 2.10)$ and the energy consumption per unit time is set as constraints (26). Two performance metrics are adopted to compare the efficiency of the two proposed methods. Hyper-volume Indicator I_h to measure the area covered by the Pareto front with a reference point equal to (1.2, 1.2). The higher value of I_h reflects a higher performance of a given method. The second indicator is the generational distance GD which provides a measure of the distance between the obtained set of solutions and the global Pareto front. See [6] for details of both metrics.

The proposed RIPG is coded with `C++` in an integrated development environment (IDE) Microsoft Visual Studio 22 while the MILP model is coded using IBM Cplex 20.0. All experiments were carried out on a Linux-based server featuring an Intel(R) Xeon(R) processor and 32 GB RAM. A stopping criteria of $CPU = (n \,.\, g \,.\, 200)$ ms is given for each instance for 10 replications. Meanwhile, for the augmented ϵ-constraint method, a total of 20 slides were used and each one is being processed within a time frame of 3 minutes (1 hour in total).

5.2. Computational Results

After conducting the experiments, the outcomes of the two methods are compared according to the considered metrics.

Table 4 shows I_h and the GD values for the small benchmark arranged by n, g and m_i. With an average $I_h = 1.23$ and $GD = 22.23$, it is clear that Aug yielded the best results for both criteria and most small instances are optimally

Table 6. I_h and GD values for Large instances (Best values are in bold)

			I_h		GD	
n	g	m	RIPG	Aug	RIPG	Aug
50	2	2	**1.15**	0.39	**3.57**	4742.44
		3	**1.28**	1.05	**10.57**	1187.20
	3	2	**0.95**	0.53	**2.64**	5797.77
		3	0.60	NA	0	NA
	4	2	**0.82**	0.29	0	1696.25
		3	0.65	NA	0	NA
			0.91	0.56	**2.80**	3355.92
100	2	2	**0.87**	0.2	**0.83**	6819.88
		3	**0.78**	0.20	**0.00**	9905.43
	3	2	**0.89**	0.19	**0.17**	13349.34
		3	0.51	NA	0	NA
	4	2	0.52	NA	0	NA
		3	0.51	NA	0	NA
			0.68	0.19	**0.17**	10024.88
Avg			**0.79**	0.41	**1.48**	6214.04

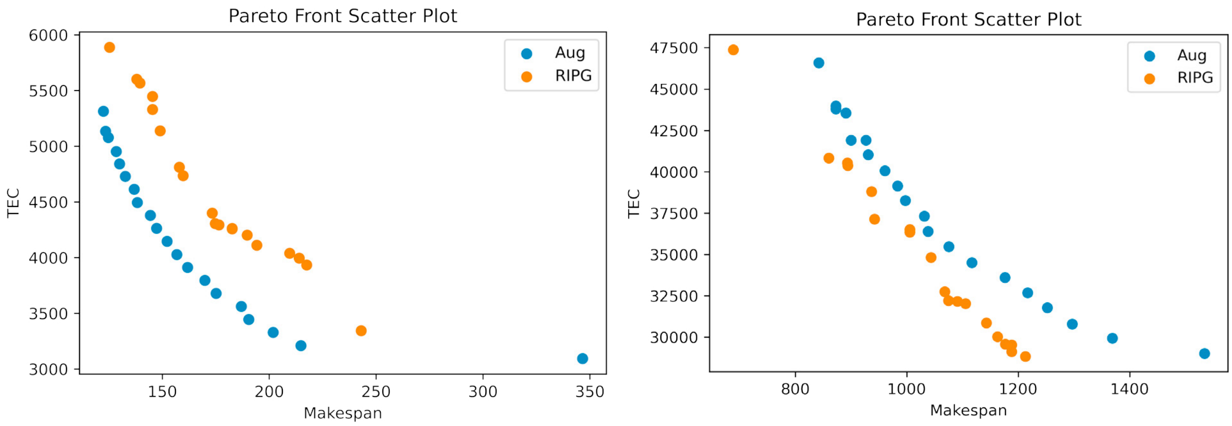

Fig. 3. Pareto fronts for successively a small and a medium instance.

solved with the MILP model. The average I_h achieved by the proposed RIPG is 0.96, a value closely approaching the optimal results, providing robust evidence of its efficacy. Table 5 presents the results of the medium benchmark including instances of $n = (15, 20, 30)$. Unlike the small set, the values of I_h returned by the two methods are very close while the GD provided by the $RIPG$ is the best which reflects the performance of our metaheuristic in terms of convergence. An example of the obtained Pareto fronts for a small instance of 6 jobs and a medium instance consisting of 30 jobs is presented in Figure 3. The results of the large benchmark are given in Table 6. As seen the MILP model is not able to solve some large instances. As a result, the proposed RIPG performs better than the Aug for large instances in both criteria.

6. Conclusion

HFS problems with different constraints and energy aspects have attracted the attention of researchers in the last decade and many works have been done aiming to find the balance between productivity and energy consumption. In this paper, the BHFS-SDST is addressed for the minimization of TEC and Makespan simultaneously. A novel MILP model is developed and the augmented ϵ constraints method is used to get the Pareto front. A new RIPG is introduced including four phases for the resolution. The computational results show the scalability and convergence of the developed heuristic. The current work could be extended by both in terms of the problem and the solution methods The addition of more practical energy aspects such as peak power and time of use energy prices can be considered. Also, the proposed RIPG could be used for the resolution of other energy-aware scheduling problems. Exploiting the developed MILP in a large neighborhood search heuristic and formulating the problem as a constraint programming model for faster exploration are promising future research directions. Finally, while the findings of this study provide valuable insights into manufacturing, they are limited by the availability of real data.

Acknowledgement

This publication was conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC and 22/NCF/DR/11264, National Challenge Fund, Digital for Resilience Challenge.

References

[1] Boukas, E.K., Haurie, A., Soumis, F., 1990. Hierarchical approach to steel production scheduling under a global energy constraint. Annals of operations research 26, 289–311.

[2] Chen, J.f., Wang, L., Peng, Z.p., 2019. A collaborative optimization algorithm for energy-efficient multi-objective distributed no-idle flow-shop scheduling. Swarm and Evolutionary Computation 50, 100557.

[3] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6, 182–197.

[4] Framinan, J.M., Leisten, R., 2008. A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. OR Spectrum 30, 787–804.

[5] Gomes, A.C.L., Ravetti, M.G., Carrano, E.G., 2021. Multi-objective matheuristic for minimization of total tardiness and energy costs in a steel industry heat treatment line. Computers & Industrial Engineering 151, 106929.

[6] He, L., Cao, Y., Li, W., Cao, J., Zhong, L., 2022. Optimization of energy-efficient open shop scheduling with an adaptive multi-objective differential evolution algorithm. Applied Soft Computing 118, 108459.

[7] Li, R., Li, J., Li, J., Duan, P., 2023. A collaboration-based multi-objective algorithm for distributed hybrid flowshop scheduling with resource constraints. Swarm and Evolutionary Computation 83, 101409.

[8] Luo, H., Du, B., Huang, G.Q., Chen, H., Li, X., 2013. Hybrid flow shop scheduling considering machine electricity consumption cost. International journal of production economics 146, 423–439.

[9] Mavrotas, G., 2009. Effective implementation of the -constraint method in multi-objective mathematical programming problems. Applied Mathematics and Computation 213, 455–465.

[10] Missaoui, A., Boujelbene, Y., 2021. Artificial bee colony for blocking flowshop with due date windows, in: 2021 International Conference on Data Analytics for Business and Industry (ICDABI), IEEE. pp. 557–561.

[11] Missaoui, A., Ozturk, C., O'Sullivan, B., 2023a. Energy-efficient multi-objective hybrid flowshop scheduling problem with blocking constraints, in: 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE. pp. 01–06.

[12] Missaoui, A., Ozturk, C., O'Sullivan, B., Garraffa, M., 2023b. Energy-Efficient Manufacturing Scheduling: A Systematic Literature Review. Springer Nature Singapore, Singapore. pp. 1–26. URL: https://doi.org/10.1007/978-981-19-8851-6_56-1, doi:10.1007/978-981-19-8851-6_56-1.

[13] Mokhtari-Moghadam, A., Pourhejazy, P., Gupta, D., 2023. Integrating sustainability into production scheduling in hybrid flow-shop environments. Environmental Science and Pollution Research , 1–19.

[14] Nawaz, M., Enscore Jr, E.E., Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega 11, 91–95.

[15] Rajendran, C., Alicke, K., 2007. Dispatching in flowshops with bottleneck machines. Computers & Industrial Engineering 52, 89–106.

[16] Ruiz, R., Şerifoğlu, F.S., Urlings, T., 2008. Modeling realistic hybrid flexible flowshop scheduling problems. Computers & Operations Research 35, 1151–1175.

[17] Ruiz, R., Vázquez-Rodríguez, J.A., 2010. The hybrid flow shop scheduling problem. European journal of operational research 205, 1–18.

[18] Schulz, S., Neufeld, J.S., Buscher, U., 2019. A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling. Journal of cleaner production 224, 421–434.

[19] Utama, D.M., Salim, A.A.P., Widodo, D.S., 2022. A novel hybrid archimedes optimization algorithm for energy-efficient hybrid flow shop scheduling. International Journal of Advances in Intelligent Informatics 8.

[20] Wang, S., Wang, X., Chu, F., Yu, J., 2020. An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. International Journal of Production Research 58, 2283–2314.

[21] Wu, S., Liu, L., 2023. Green hybrid flow shop scheduling problem considering sequence dependent setup times and transportation times. IEEE Access .

[22] Yaurima-Basaldua, V.H., Tchernykh, A., Villalobos-Rodríguez, F., Salomon-Torres, R., 2018. Hybrid flow shop with unrelated machines, setup time, and work in progress buffers for bi-objective optimization of tortilla manufacturing. Algorithms 11, 68.

[23] Zhang, B., Pan, Q.k., Gao, L., Meng, L.L., Li, X.Y., Peng, K.K., 2019. 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 50, 4984–4999.