Energy efficiency optimization with machine activation in blocking hybrid flow shop scheduling problem

1st Jinli Liu
School of Computer Science
Liaocheng University
Shandong, China
liujinli_98@163.com

2nd Haoxiang Qin
School of Computer Science
Liaocheng University
Shandong, China
987352978@qq.com

3rd Yuyan Han
School of Computer Science
Liaocheng University
Shandong, China
hanyuyan@lcu-cs.com

4th Yuting Wang
School of Computer Science
Liaocheng University
Shandong, China
wangyuting@lcu-cs.com

5th Qingda Chen
School of Electronic Information Engineering
Inner Mongolia University
Hohhot, China.
cqd0309@126.com

2025 37th Chinese Control and Decision Conference (CCDC) | 979-8-3315-1056-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/CCDC65474.2025.11090246

***Abstract*—This study tackles the blocking hybrid flow-shop scheduling problem (BHFSP), focusing on minimizing energy consumption. Unlike traditional problems, we consider machine standby, setup modes, and the absence of intermediate buffers between stages. We propose a novel mixed integer linear programming model that incorporates machine turn-on/off strategies, enhancing operational efficiency and sustainability. To solve this, we develop a problem-specific variable neighborhood search iterated greedy (VIG) algorithm, which minimizes setup and blocking times and improves solution diversity through random perturbations in job sequencing. Simulation experiments on 240 test instances show that our algorithm outperforms state-of-the-art methods, highlighting its potential for energy reduction and sustainable manufacturing practices.**

***Index Terms*—Blocking hybrid flow shop scheduling problem, machine turning on/off, energy-efficient, problem-specific iterative greedy algorithm, variable neighborhood search**

I. Introduction

With rapid technological and economic development, energy consumption and environmental pollution in manufacturing have become significant concerns. Manufacturing enterprises must reduce energy consumption to cut costs and adopt environmentally friendly practices. Current strategies mainly focus on optimizing equipment usage, though improving hardware can be costly. This paper proposes strategies to reduce energy consumption.

Scheduling, particularly flowshop scheduling (FSP), is crucial in manufacturing systems [1]. Strategies in FSP aim to reduce machine energy consumption. Mouzon et al. [2] introduced a machine on/off scheduling framework to save energy, which was validated by Che et al. [3]. The hybrid flow shop scheduling problem (HFSP), a variant of FSP, has been widely studied [4], but it is a complex optimization problem. In HFSP, jobs face blocking constraints due to insufficient intermediate buffer storage, leading to production delays.

Blocking constraints result in machine idle time, causing unnecessary energy consumption. Thus, optimizing job sequencing to minimize energy consumption is essential. However, blocking conditions vary with job sequencing, complicating the search for optimal solutions [5].

Metaheuristic algorithms have been applied to FSP, but few studies address energy consumption minimization in BHFSP. Meng et al. [6] proposed an energy-saving MILP for BHFSP but did not compare it with metaheuristics. Qin et al. [7] improved the iterated greedy (IG) algorithm for energy reduction but did not consider setup time or blocking constraints. Challenges include the impact of job sequencing on energy consumption, especially with setup time and blocking, and the difficulty of finding feasible solutions in large problems. Moreover, there is a lack of strategies to optimize both blocking constraints and machine setup time in existing metaheuristics.

To address these issues, this paper proposes an improved IG algorithm integrated with a variable neighborhood search (VIG) strategy to reduce energy consumption by optimizing job sequences and incorporating machine on/off strategies.

The main contributions are: 1) VIG algorithm to efficiently find solutions. 2) A BHFSP model with a machine on/off strategy to reduce energy consumption during machine adjustments and standby states. 3) New strategies for setup time and blocking constraints embedded into the VIG framework for better search balance.

The remainder of the paper is organized as follows: Section II reviews related research; Section III presents the MILP model for BHFSP; Section IV describes the VIG algorithm; Section V presents experiments and analysis; Section VI concludes and discusses future work.

Natural Science Foundation of Shandong province, National Natural Science Foundation of China, Guangyue Young Scholar Innovation Team of Liaocheng University

II. LITERATURE REVIEW

A. Hybrid Flow Shop Scheduling Problem

The Hybrid Flow Shop Scheduling Problem (HFSP) is NP-hard even with two stages [8]. Various metaheuristic algorithms have been proposed to solve HFSP, including those by Lin et al. [9], who considered random workstation capacities, and Meng et al. [10], who used MILP models to minimize energy consumption. Lei et al. [11] developed a teaching-learning-based optimization algorithm for energy minimization, while Pan et al. [12] proposed multiple metaheuristics to minimize makespan. Additionally, Zhang et al. [13] optimized both energy consumption and makespan using a discrete artificial bee colony algorithm. Despite these efforts, no strategies specifically address setup time and blocking constraints in HFSP. This paper focuses on these overlooked aspects, particularly energy consumption from job location and machine operation.

B. Blocking Hybrid Flow Shop Scheduling Problem

Blocking is a significant constraint in many flowshop problems, including HFSP [14]. In BHFSP, where blocking occurs due to insufficient buffer storage, previous research has focused on solving BHFSP through MILP models and metaheuristics [15]. For instance, Qin et al. [16] proposed a double-level mutation IG algorithm for BHFSP. However, these studies typically neglect energy consumption during machine setup and idle states, as well as the machine turning on/off strategy to optimize energy use. This paper addresses these gaps by proposing a novel metaheuristic algorithm to minimize energy consumption in BHFSP.

C. Iterative Greedy Algorithm

The Iterative Greedy (IG) algorithm, introduced by Ruiz [17], is a widely used metaheuristic for scheduling problems. Pan and Ruiz [18] applied IG to mixed no-idle FSP, while Wang et al. [19] optimized the mixed no-wait FSP. Several adaptations of IG for blocking constraints have also been proposed, such as by Shao et al. [20] for DFSP and Öztop et al. [21] for HFSP. However, most IG improvements focus on search strategies for job sequences, without considering energy consumption in machine setup and idle modes. To address these limitations, this paper integrates Variable Neighborhood Search (VNS) into the IG framework, resulting in the Variable Neighborhood IG (VIG) algorithm, which outperforms existing methods, particularly for larger problem sizes.

III. PROBLEM DESCRIPTION

The considered BHFSP with a machine turning on/off strategy is described as follows. The workshop has S stages, each with m machines. Each stage includes one or more parallel machines K_i = 1, 2, ..., m_i, with at least one stage having more than one machine. All machines are available at time zero. A collection of n jobs must be processed consecutively across the stages without preemption. The processing time at each stage is known in advance, and transportation time is included in the processing time. There are no buffers between adjacent stages, and jobs must wait if no machine is idle at the next stage. When a machine is in standby (blocking or idle) mode, a machine turning on/off strategy may be applied based on specific conditions. The first idle machine is selected at each stage for processing. The objective is to minimize energy consumption using the turning on/off strategy. Since the energy consumption of job processing is constant, the corresponding power is not considered. Incorporating the work of [22], the mathematical model is formulated as follows.

A. Mathematical Model

1) Notations: n: The total number of jobs.

J: Set of jobs,$J \in \{1, 2..., n\}$.

j, j': Index of the job, $j, j' \in J$.

S: The total number of stages.

I: Set of stages, $I \in \{1, 2..., S\}$.

i: Index of the stage, $i \in I$.

m: The total number of machines.

m_i: Number of parallel machines at the stage i.

K: Set of machines, $K \in \{1, 2..., m\}$.

K_i: Set of parallel machines at the stage i, $K_i \in \{1, 2..., m_i\}$.

t: Index of the machine position.

L_k, LL_k: Set of positions on machine k, $L_k \in \{1, 2..., n\}$, $LL_k \in \{1, 2..., n-1\}$.

$pt_{j,k}$: Processing time of job j on machine k.

P^k_{set}: Energy consumption of machine k at setup mode per unit time.

$Set_{j,j',k}$: Setup time from job j to j' on machine k, the first job on machine k is $Set_{j,j,k}$.

$Set_{j,j',i}$: Setup time from job j to j' at stage i, the first job on machine k is $Set_{j,j,i}$.

$Set_{k,t}$: Setup time from job t to $t+1$ on machine k.

$SE_{k,t}$: Energy consumption of machine k at time $Set_{k,t}$.

SE: Total energy consumption at setup mode.

$P^k_{standby}$: Energy consumption of machine k at standby mode per unit time.

$BIT^{k,t}_{standby}$: Blocking and idle time from position t to $t+1$ on machine k.

$WE_{k,t}$: Standby energy consumption of the machine k at time $BIT^{k,t}_{standby}$.

WE: Total energy consumption at standby mode.

$E^{on/off}_k$: Energy consumption required for one turning on/off of machine k.

$T^{on/off}_k$: Time required for one turning on/off of machine k.

$TB^{on/off}_k$: Breakeven period of machine k, i.e., the minimum waiting time if the machine can implement a turning on/off strategy..

TEC: Total energy consumption of workshop.

M: A very large positive number. 2) Decision variables: $Y_{j,k,t}$: Binary decision variable that equals 1 if job j is processed on position t of machine k; 0, otherwise.

$B_{j,i}$: Continuous decision variable, which denotes the beginning time of job j at stage i.

$E_{j,i}$: Continuous decision variable, which indicates the ending time of job j at stage i.

$D_{j,i}$: Continuous decision variable, which shows the departure time of job j at stage i.

$S_{k,t}$: Continuous decision variable denoting the starting time of position t on machine k.

$F_{k,t}$: Continuous decision variable indicating the finishing time of position t on machine k.

$ES_{k,t}$: Continuous decision variable showing the $WE_{k,t}$ from position t to $t+1$ on machine k.

$Z_{k,t}$: Binary decision variable that equals 1 if implement the turning on/off strategy from position t to $t+1$ on machine k; 0, otherwise. 3) Objective:

$$Minimize \quad TEC = SE + WE \tag{1}$$

$$Set_{k,t} = \sum_{j\in J}\sum_{j'\in J} Y_{j,k,t}Y_{j',k,t+1}Set_{j,j',k} \tag{2}$$

$$\begin{aligned} SE &= \sum_{k\in K}\sum_{t\in L_k} SE_{k,t} = \sum_{k\in K}\sum_{t\in L_k} Set_{k,t}P^k_{set} \\ &= \sum_{k\in K}\sum_{t\in LL_k}\sum_{j\in J}\sum_{j'\in J} Y_{j,k,t}Y_{j',k,t+1}Set_{j,j',k}P^k_{set} \end{aligned} \tag{3}$$

$$BIT^{k,t}_{standby} = S_{k,t+1} - F_{k,t} - Set_{k,t} \tag{4}$$

$$TB^{on/off}_k = max\{T^{on/off}_k, E^{on/off}_k / P^k_{standby}\} \tag{5}$$

$$\begin{aligned} WE &= \sum_{k\in K}\sum_{t\in LL_k} WE_{k,t} = \sum_{k\in K}\sum_{t\in LL_k} \\ &((1 - Z_{k,t})BIT^{k,t}_{standby}P^k_{standby} + Z_{k,t}E^{on/off}_k) \end{aligned} \tag{6}$$

s.t.

$$\sum_{k\in K}\sum_{t\in L_k} Y_{j,k,t} = 1, \forall i \in I, j \in J \tag{7}$$

$$\sum_{j\in J} Y_{j,k,t} \geqslant \sum_{j\in J} Y_{j,k,t+1}, \forall k \in K, t \in LL_k \tag{8}$$

$$\sum_{j\in J} Y_{j,k,t} \leqslant 1, \forall k \in K, t \in L_k \tag{9}$$

$$D_{j,i} = B_{j,i+1}, \forall j \in J, i \in \{1, 2..., S-1\} \tag{10}$$

$$E_{j,i} = B_{j,i} + \sum_{k\in K}\sum_{t\in L_k} pt_{j,k}Y_{j,k,t}, \forall j \in J, i \in I \tag{11}$$

$$F_{k,t} = S_{k,t} + \sum_{j\in J} pt_{j,k}Y_{j,k,t}, \forall k \in K, t \in L_k \tag{12}$$

$$E_{j,i} \leqslant D_{j,i}, \forall i \in I, j \in J \tag{13}$$

$$\begin{aligned} &B_{j',i} \geqslant D_{j,i} + Set_{j,j',k} - M(2 - Y_{j,k,t} - Y_{j',k,t+1}) \\ &\forall j, j' \in J, j \neq j', i \in I, k \in K_i, t \in LL_k \end{aligned} \tag{14}$$

$$S_{k,t} \leqslant B_{j,i} + M(1 - Y_{j,k,t}), \forall i \in I, j \in J, k \in K_i, t \in L_k \tag{15}$$

$$S_{k,t} \geqslant B_{j,i} - M(1 - Y_{j,k,t}), \forall i \in I, j \in J, k \in K_i, t \in L_k \tag{16}$$

$$\begin{aligned} &TB^{on/off}_k Z_{k,t} + Set_{j,j',k} \leqslant \\ &S_{k,t+1} - F_{k,t} + M(2 - Y_{j,k,t} - Y_{j',k,t+1}) \\ &\forall j, j' \in J, j \neq j', i \in I, k \in K, t \in LL_k \end{aligned} \tag{17}$$

$$\begin{aligned} &E^{on/off}_k Z_{k,t} + P^k_{set}Set_{j,j',k} \leqslant \\ &ES_{k,t} + M(2 - Y_{j,k,t} - Y_{j',k,t+1}) \\ &\forall j, j' \in J, j \neq j', i \in I, k \in K, t \in LL_k \end{aligned} \tag{18}$$

$$\begin{aligned} &(S_{k,t+1} - F_{k,t} - Set_{j,j',k})P^k_{standby} + P^k_{set}Set_{j,j',k} \leqslant \\ &ES_{k,t} + MZ_{k,t} + M(2 - Y_{j,k,t} - Y_{j',k,t+1}), \\ &\forall j, j' \in J, j \neq j', i \in I, k \in K, t \in LL_k \end{aligned} \tag{19}$$

$$F_{k,t} \leqslant S_{k,t+1}, \forall k \in K, t \in LL_k \tag{20}$$

$$0 \leqslant ES_{k,t}, \forall k \in K, t \in LL_k \tag{21}$$

$$B_{j,i}, S_{k,t} \geqslant 0, \forall j \in J, k \in K, i \in I_j, t \in L_k \tag{22}$$

The optimization objective is defined in Equation (1). The total energy consumption is the sum of setup and standby modes. Equation (2) ensures that $Set_{k,t}$ is determined by the procedure from t to $t+1$ positions. Equation (3) calculates setup energy consumption. Equation (4) computes the machine's standby time, which includes blocking and idle time. Equation (5) determines the machine's breakeven period. Equation (6) defines final standby energy consumption as the sum of standby and turning on/off energy. Constraint (7) ensures each job is assigned to one machine. Constraint (8) arranges processing according to position sequence. Constraint (9) limits a position to process at most one job at a time. Constraint (10) enforces blocking, i.e., the departure time of a job from the previous stage equals its start time in the next stage. Constraint (11) calculates the job's end time as the sum of start time and processing time. Constraint (12) ensures the machine's end time equals the sum of its start time and the processing time. Constraint (13) ensures a job's end time is not greater than its exit time from the machine. Constraint (14) requires the machine's last position start time to be at least the sum of departure and setup times from the previous position. Constraints (15) and (16) ensure that the job's start time at a machine position equals the machine's start time. Constraints (17-19) relate time and energy consumption during the on/off strategy. Constraint (20) ensures the finishing time of the previous machine position does not exceed the start time of the next. Constraints (21-22) limit the decision variables' range.

IV. Proposed Algorithm

A. The framework of the VIG algorithm

The VIG algorithm consists of three main components: Initialization, Destruction-Reconstruction, and the Variable Neighborhood Search (VNS). The VNS incorporates two operators based on blocking constraints and setup time to optimize the job sequence and improve the algorithm's intensity. To enhance solution diversity, a random perturbation strategy is added.

B. Initialization strategy

The total energy consumption in a workshop is closely related to the job sequence and schedule. A well-optimized sequence can reduce machine energy use in standby and setup modes. In the initialization stage, the NEH heuristic [23] is applied to reorder jobs based on their setup times. The sequence with the lowest energy consumption is selected iteratively, as shown in Algorithm 1.

Algorithm 1 NEH initialization strategy

Require: π^{origin}
Ensure: π
1: $\pi^{\text{temp}} \leftarrow$ *Sort_descending*$(\sum_{k=1}^{S} pt_{j,k})$, *j=1,2,..,n*
2: **for** $j = 1 \rightarrow n$ **do**
3: $\quad \pi \xleftarrow[i=1\ to\ |\pi|]{insert\ ith\ posistion}$ *extract*(π^{temp}_j)
4: $\quad \pi \leftarrow argmin^{|\pi|}_{i=1} f(\pi)$
5: **end for**

C. Destruction-Reconstruction strategy

This strategy aims to minimize energy consumption by altering job positions. A set of jobs is randomly selected and reinserted into the sequence, retaining the configuration with the lowest energy consumption. This strategy deepens the local search and intensifies the algorithm's exploration. The process is detailed in Algorithm 2.

Algorithm 2 Destruction-Reconstruction strategy

Require: π, *parameter d, unscheduled sequence Set* $U_{i=1}^{d}$
Ensure: π^{temp}
$\pi^{\text{temp}} = \pi$
$U_{i=1}^{d}$= *extract*(π^{temp})
for $j = 1 \to d$ **do**
    $\pi^{\text{temp}'} \leftarrow \pi^{\text{temp}} \backslash U_j$
    $\pi^{\text{temp}'} \xleftarrow[i=1 \;\; to \;\; |\pi^{\text{temp}'}|]{insert \;\; ith \;\; posistion} U_j$
    $\pi^{\text{temp}'} \leftarrow argmin_{i=1}^{|\pi^{\text{temp}'}|}$ *f*($\pi^{\text{temp}'}$)
end for
if f($\pi^{\text{temp}'}$) $<$ f(π^{temp}) **then**
    $\pi^{\text{temp}} = \pi^{\text{temp}'}$
end if

Algorithm 3 VNSs

Require: π^{temp}, *N*, $Set_{j,j',i}$
Ensure: π^{temp}
for $c = 1 \to N$ **do**
    Max_Setup = $Set_{\pi_1,\pi_1,1}$
    $Pos = 1$
    for $j = 1 \to |\pi|$-1 **do**
        if *Max_Setup* $< Set_{\pi_j,\pi_{j'},1}$ **then**
            Max_Setup = $Set_{\pi_j,\pi_{j'},1}$
            $Pos = j'$
        end if
    end for
    for $j = 1 \to |\pi|$ **do**
        if $j \neq Pos$ **then**
            $\pi^{\text{temp}'} \leftarrow$ *Swap*(π_j,π_{Pos})
            if f($\pi^{\text{temp}'}$) $<$ f(π^{temp}) **then**
                $\pi^{\text{temp}} = \pi^{\text{temp}'}$
            else
                $\pi^{\text{temp}'} = \pi^{\text{temp}}$
            end if
        end if
    end for
end for

D. Problem-Specific Variable Neighborhood Search strategy

A problem-specific VNS strategy can directly reduce the energy consumption of the scheduling. Two operators are designed according to the setup time and blocking conditions of jobs, respectively. The operator aims at setup time, named VNS$_S$, selecting the job with the maximum setup time to swap with all other jobs. After each exchange, if the energy consumption value decreases, the current scheduling sequence will be retained. Similar to the VNS$_S$ strategy, for blocking constraints, the VNS$_B$ operator is designed to swap the maximum blocking job with other jobs. If the energy consumption

Algorithm 4 VNS$_B$

Require: π^{temp}, *N*, $Blocktime_{j,i}$, % Record the blocking time of job on position *j* at stage *i*
Ensure: π^{temp}
for $c = 1 \to N$ **do**
    Max_Blocking = 0
    $Pos = 1$
    for $j = 1 \to |\pi|$-1 **do**
        if *Max_Blocking* $< Blocktime_{j,i}$ **then**
            Max_Blocking = $Blocktime_{j,i}$
            $Pos = j$
        end if
    end for
    for $j = 1 \to |\pi|$ **do**
        if $j \neq Pos$ **then**
            $\pi^{\text{temp}'} \leftarrow$ *Swap*(π_j,π_{Pos})
            if f($\pi^{\text{temp}'}$) $<$ f(π^{temp}) **then**
                $\pi^{\text{temp}} = \pi^{\text{temp}'}$
            else
                $\pi^{\text{temp}'} = \pi^{\text{temp}}$
            end if
        end if
    end for
end for

Algorithm 5 VNS$_R$

Require: π^{temp}, *P*
Ensure: π^{temp}
for $c = 1 \to P$ **do**
    $pt1 = pt2 = 0$
    while $pt1 == pt2$ **do**
        $pt1 = rand()\%n$
        $pt2 = rand()\%n$
    end while
    $\pi^{\text{temp}'} \leftarrow$ *Swap*(π_j,π_{Pos})
    if f($\pi^{\text{temp}'}$) $<$ f(π^{temp}) **then**
        $\pi^{\text{temp}} = \pi^{\text{temp}'}$
    else
        $\pi^{\text{temp}'} = \pi^{\text{temp}}$
    end if
end for

of the job sequence can be reduced, the new sequence replaces the old one. In addition, to balance the intensification and diversification of the algorithm, further improving its global search capability, a random operator VNS$_R$ is utilized to perturb the job sequence to prevent the solution from falling into local optimization. In the VNS, greedy minds are applied, all operators are executed one by one, and each operator must be executed *C* times. However, once the algorithm uses the operator to improve the incumbent sequence, the operator will be executed for the first time. Only when the current operator is not improved after *C* iterations, other operators can be executed. When executing these three operators, the machine turning on/off strategy is implemented to help the workshop reduce the energy consumption of the standby mode. Once the machine is at idle or blocking state, and its duration exceeds the specific threshold, this machine turning on/off strategy will be enabled. The three operators are shown in Algorithms 3, 4, and 5 respectively, and the framework of the Problem-Specific

VNS strategy is given in Algorithm 6.

Algorithm 6 Problem-Specific VNS

Require: π^{temp}, *num, C, N, P*
Ensure: π
while $u \leqslant num$ **do**
  while $count \leqslant C$ **do**
    Problem Specific VNS strategy:
    Case1: $u == 1$
    $\pi^{\text{temp}'} \leftarrow VNS_R(\pi^{temp}, P)$
    Case2: $u == 2$
    $\pi^{\text{temp}'} \leftarrow VNS_S(\pi^{temp}, N)$
    Case3: $u == 3$
    $\pi^{\text{temp}'} \leftarrow VNS_B(\pi^{temp}, N)$
    if $f(\pi^{\text{temp}'}) < f(\pi^{\text{temp}})$ **then**
      $\pi^{\text{temp}} = \pi^{\text{temp}'}$, $count = 0$
    else
      $\pi^{\text{temp}} = \pi^{\text{temp}}$, $count++$
    end if
  end while
  $u++$
end while
if $f(\pi^{\text{temp}}) < f(\pi)$ **then**
  $\pi = \pi^{\text{temp}}$
end if

V. COMPUTATIONAL RESULTS

A. Experimental conditions and evaluation indicators

We first investigate the parameters of the proposed algorithm and test the effectiveness of VNS, and then compare the VIG with classic and state-of-the-art metaheuristics. In literature [14], we consider the scale $n \in \{10, 20, 30, 40, 50, 60, 70, 80\}$ and set $S \in \{5, 8, 10\}$, respectively, and the number of identical parallel machines m_i in each stage are randomly generated between intervals [1,3]. For each combination *(n, i)*, ten instances are yielded. Therefore, the number of the combinations is 8×3×10 = 240. According to the literature [14], For all types of instances, the set of setup times should not exceed 20% of the processing times. It generally represents wider real-world manufacturing cases. Furthermore, on the basis of data sets in literature [10], we set the processing time, $pt_{j,k}$ between [20, 60] and $Set_{j,j',i}$ between [4, 12], which obey the uniform distribution. The power consumption of the standby $P^k_{standby}$ and setup mode P^k_{set} of machines are randomly generated in intervals [1, 3], [12, 24]. The energy consumption $E^{on/off}$ and time $T^{on/off}$ required for one turning on/off of machine *k* are uniformly sampled from the set [10, 60] and [8, 16].

The comparison and analysis among of different parameters, components, and algorithms are implemented in different instances of the problem. We first compare the results by using the relative percentage increase (RPI) [24], which is calculated as a response variable and shown in Equation (23).

$$RPI = (c_i - c_{best})/c_{best} \times 100 \tag{23}$$

Where c_i is the total energy consumption obtained by algorithm *i* and c_{best} is the best value obtained by all algorithms. Refer to the algorithm time setting mode in paper [25], the maximum elapsed CPU running time is set to $T_{max} = \tau \times n \times S$ milliseconds, where τ is a parameter and is set with two values: 5, 10 to provide an overall performance of all the algorithms.

B. Effectiveness of the VIG algorithm

To the best of our knowledge, the IGS algorithm [7] is the only existing metaheuristic for solving the BHFSP with energy minimization. Thus, we use IGS as a comparative algorithm. Additionally, we evaluate classical metaheuristics such as GA, Simulated Annealing (SA) [10], and Discrete Particle Swarm Optimization (DPSO) [26], which address energy consumption in the HFSP with and without the machine on/off strategy. And the Double Level Mutation IG (IGDLM) algorithm [16], which minimizes makespan for the BHFSP. For a fair comparison, all algorithms are reimplemented based on their original settings, and the machine on/off strategy is embedded. All experiments are conducted under the same conditions with the same termination time, and $\tau = 10$. We test each algorithm across all instances with the machine on/off strategy. The smaller ARPI indicate better overall performance. Fig. 1 presents the 95% Tukey's HSD confidence intervals. When the on/off strategy is applied, the gap between DPSO, IGDLM, IGS, and VIG narrows, but VIG shows a significant advantage, confirming the effectiveness of VIG, particularly.

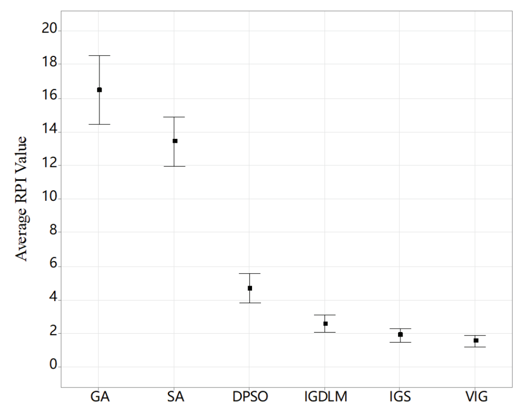

Fig. 1. Comparisons for all algorithms with machine turning on/off strategy when $\tau = 10$.

VI. CONCLUSION

This paper investigated a blocking hybrid flow shop scheduling problem with a machine turning on/off strategy for minimizing energy consumption and established BHFSP mathematical model. To solve this problem, iterated greedy algorithm with the VNS strategy is proposed to solve this problem. In the proposed VIG algorithm, new strategies are developed and embedded into the problem-specific VNS framework to adjust the blocking condition of jobs and machine setup time. We compared the performance of different algorithms in the same environment. A large number of simulation experiments and statistical analyses show that the VIG algorithm is efficient for solving the problem.

In the future, we will further design specific strategies according to practical needs and problems encountered. It is interesting to design various efficient optimization operators for different constraints, such as deteriorating jobs, batch-processing machines, dynamic operation skipping, and so on. The IG algorithm is a potential algorithm because of its simple structure, few parameters, and easy implementation. We will try our best to improve its performance. Simultaneously, the multiple objectives of the simultaneous optimization problem will be considered, such as the makespan, machine running speed, energy consumption, total flow time, and so on.

A. Authors and Affiliations

Jinli Liu Liaocheng University, Liaocheng, Shandong, China. **Haoxiang Qin** Liaocheng University, Liaocheng, Shandong, China. **Yuyan Han** Liaocheng University, Liaocheng, Shandong, China. **Yuting Wang** Liaocheng University, Liaocheng, Shandong, China. **Qingda Chen** School of Electronic Information Engineering, Inner Mongolia University, Hohhot, China.

ACKNOWLEDGMENT

This work was jointly supported by the Natural Science Foundation of Shandong province under grant numbers ZR2023MF022, National Natural Science Foundation of China under grant numbers 61973203, 61803192, 62106073, and 61966012. We are grateful for Guangyue Young Scholar Innovation Team of Liaocheng University under grant number LCUGYTD2022-03.

REFERENCES

[1] Fuqing Zhao, Xuan He, and Ling Wang. A Two-Stage Cooperative Evolutionary Algorithm With Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem. *IEEE Transactions on Cybernetics*, 51(11):5291–5303, November 2021. Conference Name: IEEE Transactions on Cybernetics.

[2] Gilles Mouzon, Mehmet B. Yildirim, and Janet Twomey. Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 45(18-19):4247–4271, September 2007. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207540701450013.

[3] Ada Che, Xueqi Wu, Jing Peng, and Pengyu Yan. Energy-efficient bi-objective single-machine scheduling with power-down mechanism. *Computers & Operations Research*, 85:172–183, September 2017.

[4] Junqing Li, Hongyan Sang, Yuyan Han, Cungang Wang, and Kaizhou Gao. Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production*, 181:584–598, April 2018.

[5] Yuyan Han, Dunwei Gong, Yaochu Jin, and Quanke Pan. Evolutionary Multiobjective Blocking Lot-Streaming Flow Shop Scheduling With Machine Breakdowns. *IEEE Transactions on Cybernetics*, 49(1):184–197, 2019. Conference Name: IEEE Transactions on Cybernetics.

[6] L. L. Meng, C. Zhang, B. Zhang, and J. Q. Li. Modeling of energy-saving blocking hybrid flow shop scheduling problem. *Journal of Huazhong University of science and technology*, 49(7):127–132, 2021.

[7] Haoxiang Qin, Yuyan Han, Biao Zhang, Leilei Meng, Yiping Liu, Quanke Pan, and Dunwei Gong. An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem. *Swarm and Evolutionary Computation*, 69:100992, March 2022.

[8] Jatinder N. D. Gupta. Two-Stage, Hybrid Flowshop Scheduling Problem. *Journal of the Operational Research Society*, 39(4):359–364, April 1988.

[9] Yikuei Lin, Dinghsiang Huang, and Louis Chenglu Yeng. Reliability Evaluation of a Hybrid Flow-Shop With Stochastic Capacity Within a Time Constraint. *IEEE Transactions on Reliability*, 65(2):867–877, June 2016. Conference Name: IEEE Transactions on Reliability.

[10] Leilei Meng, Chaoyong Zhang, Xinyu Shao, Yaping Ren, and Caile Ren. Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. *International Journal of Production Research*, 57(4):1119–1145, February 2019. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2018.1501166.

[11] Deming Lei, Liang Gao, and Youlian Zheng. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop. *IEEE Transactions on Engineering Management*, 65(2):330–340, May 2018. Conference Name: IEEE Transactions on Engineering Management.

[12] Quanke Pan, Liang Gao, Xinyu Li, and Kaizhou Gao. Effective meta-heuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Applied Mathematics and Computation*, 303:89–112, June 2017.

[13] Biao Zhang, Quanke Pan, Liang Gao, Xinyu Li, Leilei Meng, and Kunkun Peng. A multiobjective evolutionary algorithm based on decomposition for hybrid flowshop green scheduling problem. *Computers & Industrial Engineering*, 136:325–344, October 2019.

[14] Said Aqil and Karam Allali. Two efficient nature inspired meta-heuristics solving blocking hybrid flow shop manufacturing problem. *Engineering Applications of Artificial Intelligence*, 100:104196, April 2021.

[15] Leilei Meng, Chaoyong Zhang, Caile Ren, Zhenguo Li, and Yaping Ren. MILP Models and an Improved BSA for Hybrid Flow Shop Scheduling Problems with Blocking. *China Mechanical Engineering*, 29(22):2647–2658, November 2018. Number: 22.

[16] Haoxiang Qin, Yuyan Han, Qingda Chen, Junqing Li, and Hongyan Sang. A Double Level Mutation Iterated Greedy Algorithm for Blocking Hybrid Flow Shop Scheduling. *Control and Decision*, 37(9):2323–2332, 2022.

[17] Rubén Ruiz and Thomas Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, March 2007.

[18] Quan-Ke Pan and Rubén Ruiz. An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44:41–50, April 2014.

[19] Yamin Wang, Xiaoping Li, Rubén Ruiz, and Shaochun Sui. An Iterated Greedy Heuristic for Mixed No-Wait Flowshop Problems. *IEEE Transactions on Cybernetics*, 48(5):1553–1566, May 2018. Conference Name: IEEE Transactions on Cybernetics.

[20] Zhongshi Shao, Weishi Shao, and Dechang Pi. Effective constructive heuristic and iterated greedy algorithm for distributed mixed blocking permutation flow-shop scheduling problem. *Knowledge-Based Systems*, 221:106959, June 2021.

[21] Hande Öztop, M. Fatih Tasgetiren, Deniz Türsel Eliiyi, and Quanke Pan. Metaheuristic algorithms for the hybrid flowshop scheduling problem. *Computers & Operations Research*, 111:177–196, November 2019.

[22] Leilei Meng, Kaizhou Gao, Yaping Ren, Biao Zhang, Hongyan Sang, and Zhang Chaoyong. Novel MILP and CP models for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. *Swarm and Evolutionary Computation*, 71:101058, June 2022.

[23] Muhammad Nawaz, E Emory Enscore, and Inyong Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95, January 1983.

[24] Quanke Pan, Liang Gao, and Ling Wang. An Effective Cooperative Co-Evolutionary Algorithm for Distributed Flowshop Group Scheduling Problems. *IEEE Transactions on Cybernetics*, 52(7):5999–6012, July 2022. Conference Name: IEEE Transactions on Cybernetics.

[25] Biao Zhang, Quanke Pan, Leilei Meng, Xinli Zhang, Yaping Ren, Junqing Li, and Xuchu Jiang. A collaborative variable neighborhood descent algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Applied Soft Computing*, 106:107305, July 2021.

[26] M.K. Marichelvam, M. Geetha, and Ömür Tosun. An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study. *Computers & Operations Research*, 114:104812, February 2020.