❒ 1154

An energy-efficient flow shop scheduling using hybrid Harris hawks optimization

Dana Marsetiya Utama[1], Dian Setiya Widodo[2]
[1]Department of Industrial Engineering, University of Muhammadiyah Malang, Indonesia
[2]Department of Manufacturing Technology, Vocational Faculty, University of 17 Agustus 1945 Surabaya, Indonesia

Article Info

ABSTRACT

The energy crisis has become an environmental problem, and this has received much attention from researchers. The manufacturing sector is the most significant contributor to energy consumption in the world. One of the significant efforts made in the manufacturing industry to reduce energy consumption is through proper scheduling. Energy-efficient scheduling (EES) is a problem in scheduling to reduce energy consumption. One of the EES problems is in a flow shop scheduling problem (FSSP). This article intends to develop a new approach to solving an EES in the FSSP problem. Hybrid Harris hawks optimization (hybrid HHO) algorithm is offered to resolve the EES issue on FSSP by considering the sequence-dependent setup. Swap and flip procedures are suggested to improve HHO performance. Furthermore, several procedures were used as a comparison to assess hybrid HHO performance. Ten tests were exercised to exhibit the hybrid HHO accomplishment. Based on numerical experimental results, hybrid HHO can solve EES problems. Furthermore, HHO was proven more competitive than other algorithms.

Corresponding Author:

Dana Marsetiya Utama
Department of Industrial Engineering
University of Muhammadiyah Malang
Jl. Raya Tlogomas 246 Malang, 65144, Indonesia
Email: dana@umm.ac.id

1. INTRODUCTION

Recently, an environmental problem has received much attention from researchers. One of those problems is the energy crisis. The consumption of non-renewable energy causes the depletion of reserves of energy resources. At present, the energy supply is dominated by fossil fuels [1]. The total world energy use half is consumed in the industrial sector [2]. The industry of manufacturing is the biggest consumer of fuel energy. Energy consumption is dominated by non-renewable energy, which causes climate change. Therefore, to reduce energy consumption, manufacturing companies also must involve [2]. Increased energy consumption and energy crisis encourage manufacturing companies to minimize energy consumption during the production process [3]. Generally, during the manufacturing process, energy consumption is large. Idle time consumes most of the energy in the production process [4]. Therefore, proper scheduling is proposed to solve this problem to reduce energy consumption. Researchers call this problem energy-efficient scheduling (EES) [5]. EES is a scheduling problem that focuses on energy-efficient in the manufacturing process [6]. EES has received attention in recent years because it can reduce energy consumption without investment costs [7].

Flow shop scheduling problem (FSSP) is one of the exciting issues in the EES. This problem has an n-job that is done at the m-stage in the same order [8]. Garey, *et al*. [9] claimed that the FSSP problem could not be solved in polynomial time if the number of machines was more than 2. Therefore, FSSP is categorized as the NP-Hard problem [9, 10]. Several sophisticated algorithms have been proposed to resolve EES on FSSP. Prior study shows that the heuristic and metaheuristic procedure has attracted many researchers [11]. Some of these algorithms include the genetic algorithm (GA) [12], Nahwas Enscore Ham (NEH) algorithm [13], Campble Dudeck Smith (CDS) [14], ant colony optimization (ACO) [15], and Hybrid Metaheuristic [10]. While several studies have been conducted on the EES problem at FSSP, no studies have been found that consider the sequence-dependent setup. Setup time has an essential role in scheduling [16]. The sequence job also affects idle time [17]. High idle time increases energy consumption. Therefore, the right sequence can reduce energy consumption [18].

Over the past few years, flow shop scheduling problems have become an essential topic for energy efficiency. To our knowledge, there are no studies that discuss EES on FSSP by considering the sequence-dependent setup using hybrid Harris hawks optimization (HHO). HHO algorithm is a new algorithm of metaheuristic that imitates Harris hawks animals' prey hunting behavior [19]. Heidari, *et al*. [19] were researchers that introduced the HHO algorithm to solve a continuous optimization problem. This procedure effectively employed to complete the issue of the photovoltaic models [20], [21], manufacturing operation [22], [23], and color image segmentation [24]. These are our motives for using HHO to solve the EES on the FSSP problem. Therefore, this article aims to develop a new approach to solving an energy-efficient scheduling (EES) in FSSP with the sequence-dependent setup. This research offers a hybrid Harris hawks optimization (hybrid HHO) algorithm. This study carries out numerical tests and comparisons by the state-of-the-art algorithm to test the algorithm performance. This research has two main contributes; 1) In to the EES field on FSSP, hybrid HHO is a new competitive algorithm for solving EES problems in flow shop scheduling with the sequence-dependent setup; and 2) this research investigated the parameter influence in the EES problems.

2. PROPOSED ALGORITHM

We proposed hybrid HHO, which combined HHO with strategies swap and flip in the stage local search. The exploration and the exploitation phase are two main strategies for the HHO algorithm in search of prey. Hard besiege, soft besiege, hard besiege with progressive rapid lived, and soft besiege with progressive rapid lived are four main behaviors of the HHO algorithm. These behaviors are in the exploitation phase. This algorithm was used to solve continuous problems. Therefore, we developed the HHO algorithm to minimize energy consumption in FSSP with the sequence-dependent setup time. FSSP was considered as an NP-hard combinatorial problem that can be solved by discrete search space. We propose four stages in the Hybrid HHO algorithm. These stage includes; 1) The random population of rabbits and large rank value (LRV), 2) Exploration phase, 3). Exploitation phase, and 4) Local search phase with flip and swap. Pseudocode hybrid HHO (proposed procedure) is shown in Algorithm 1. A description of the stages of the Hybrid HHO algorithm is explained in the following section.

```
Algorithm 1 Pseudocode of Hybrid HHO algorithm
Inputs: Maximum iterations t and the population size N
Outputs: fitness value and the rabbit location
random population initialization Xi(i = 1,2,...,N)
for iteration 1 to t
fitness value calculation of hawks = X*
Set Xrabbit as the rabbit location (best location)
        for (each hawk (Xi)) do
        Reform the first energy E0 and jump strength J (E0=2rand()-1, J=2(1-rand())
        Update the E using (3))
        if (|E|≥ 1) then (Exploration phase)
                The location vector update using (1)
         if (|E| < 1) then (Exploitation phase)
                if (r ≥0.5 and |E|≥ 0.5 )then (The vector location update of soft
        besiege using (4))
                else if (r ≥0.5 and |E| < 0.5 ) then (The location vector update of hard
                besiege using (6))
                else if (r <0.5 and |E|≥ 0.5 ) then (The location vector update of soft
                besiege with progressive rapid dives using (20)
                else if (r <0.5 and |E| < 0.5 ) then (The location vector update of hard
                besiege with progressive rapid dives using (10)
        assure no repetitive number in the same s rabbit position x^{t+1}
        Apply LRV on each rabbit position x^{t+1}
        for i = 0:n
```

```
            Perform swap on the rabbit position x^{t+1}
            if (evaluate (x^{t+1}) < evaluate (X*))
                X* = x^{t+1}
            end if
        end for
        do a flip operation on a random rabbit position x_r
        for i = 0:n
        if (evaluate (x_r) < evaluate (X*))
            X* = x_r
        end if
        end for
        X* = Perform local search strategy on the best rabbit position X*
        Evaluate the rabbit position
        Update X* if there is a better solution in the population
Return Xrabbit
```

2.1. The random population of rabbit and large rank value

The position of the initial rabbit population was generated randomly. This research proposed that population positions had to ensure that no numbers were repeated. Figure 1 shows the illustration of the initialization of the rabbits' random population. In Figure 1 there were the same numbers in the population. It resulted in a population that could not be converted into a discrete search space. In converting continuous numbers to discrete search space, this research proposed converting rabbit positions (continuous value) to job permutations by applying LRV. LRV is seen as a successful way to convert continuous values to job sequence of permutation [25]. In LRV, continuous values were ranked from the largest to the smallest values. Figure 2 describes the Illustration of the LRV application.

$$\begin{bmatrix} 0.12 & 0.29 & 0.11 & 0.81 \\ 0.61 & 0.43 & 0.88 & 0.76 \\ 0.55 & 0.72 & 0.72 & 0.94 \end{bmatrix} \qquad\qquad \begin{bmatrix} 0.12 & 0.33 & 0.81 & 0.81 \\ 0.61 & 0.82 & 0.43 & 0.43 \\ 0.55 & 0.76 & 0.94 & 0.94 \end{bmatrix}$$

Accepted population   *Rejected population*

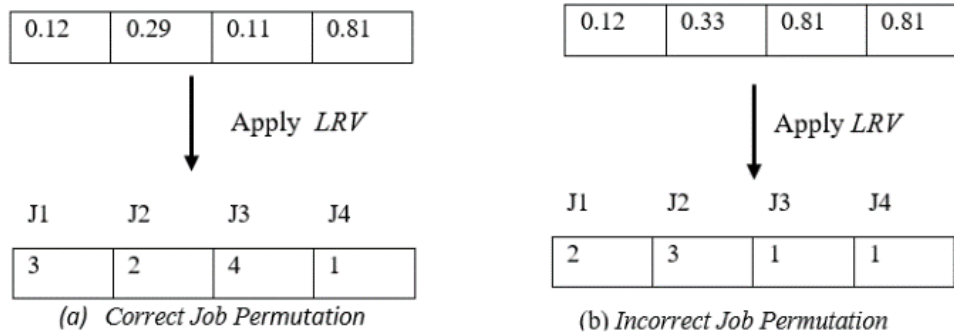Figure 1. Illustration of rabbit random population initiation

Figure 2. Illustration of LRV application

2.2. Exploration phase

In this phase, Harris hawk perched randomly in several locations. They were waiting for their rabbit prey. This behavior is modeled in (1):

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{1}$$

X (t+1) showed the vector position of the eagle in the further iteration. The current eagle position vector is depicted as X (t). X_rabbit (t) indicated the rabbit position. UB and LB projected the upper bound and lower bound variable limits. r1, r2, r3, r4, and q were random values in the range (0.1). This number was updated in

every iteration. Xrand (t) formulated a randomly picked eagle of the current population. $X_m(t)$ described the average position of the current eagle population. $X_m(t)$ was formulated with (2).

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{2}$$

$X_i(t)$ was the location of each eagle in the iteration t. Furthermore, N was the total number of eagles. The HHO algorithm was transferred from exploration to exploitation. That behavior was based on prey energy (rabbit). The energy of prey (rabbit) was drastically reduced during escape behavior. In (3) showed the prey energy formula (rabbit).

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{3}$$

E indicated the energy coming out of the prey. T formulated the maximum number of iterations. $2E_0$ displayed the initial state of energy of the rabbit.

2.3. Exploitation phase

In this section, the exploitation phase strategies of HHO are explained. Harris hawk performed four strategies: hard besiege, soft besiege, hard besiege with progressive rapid dives, and soft besiege with progressive rapid dives. In the soft besiege strategy, when the rabbits still had enough energy, they tried to escape from Harris hawk. These eagles then surrounded it gently, and they made the rabbit more tired. Furthermore, they did a surprise pounce. This behavior was formulated in (4) and (5).

$$X(t + 1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{5}$$

ΔX (t) showed the difference between the rabbit position vector and the current location in iteration t. r5 described random numbers in the range (0.1). During the runaway procedure, the rabbit random jump strength was formulated as J=2(1-r5). The value of J changed randomly at each iteration. The second strategy was hard besieged. Rabbits were already exhausted, and they had low runaway energy. Harris hawk did not surround rabbits, and they did pounce shocks. A new position on the hard besiege strategy was formulated in (6).

$$X(t + 1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{6}$$

The third strategy was soft besiege with progressive rapid dives. Rabbits were assumed to have enough energy to escape successfully because Harris hawk built a soft siege before the shock pounced. To make a soft siege, the eagle could evaluate the next step based on (7). The current position was updated using (6). In this strategy, Harris hawks did irregular, sudden, and fast dives to get close to its prey. Harris hawks dive pattern was formulated in (8).

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{7}$$

$$Z = Y + S \; x \; LF(D) \tag{8}$$

LF indicated the levy fight function. S showed a random vector of size 1xD. D described the dimensions of the problem. The LF function was calculated using (9).

$$LF(x) = 0.01 \; x \; \frac{ux\sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta)xsinsin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)x\,\beta\,x\,2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{9}$$

U and v were random values in the range (0.1), β was the default constant of 1.5. Therefore, the strategy of Soft besieges with progressive rapid dives was shown in (10). Y and Z were obtained using (7) and (8).

$$X(t + 1) = \{Y \; if \; F(Y) < F(X(t)) \; Z \; if \; F(Z) < F(X(t)) \tag{10}$$

The fourth strategy was hard to besiege with progressive rapid dives. This behavior was carried out if the rabbit did not have enough energy to escape. Harris hawk does hard besiege. Furthermore, they surprise shocking rabbits. In (10) shows the rule in hard besiege conditions. Y and Z were obtained using the new rules in (7) and (8). $X_m(t)$ was obtained by using (2).

2.4. Local search

The swap and flip are two are local search procedures proposed to improve the HHO performance. The swap illustration can is seen in Figure 3. In the swap operation, two positions were chosen randomly, and they were then swapped. Furthermore, the flip procedure was done by reversing the randomly selected job sequence. The flip operation was illustrated in Figure 4. In hybrid HHO, each iteration t, swap, and flip operations were repeated by the number of jobs.
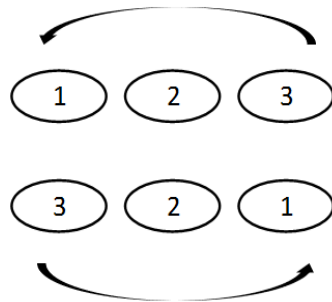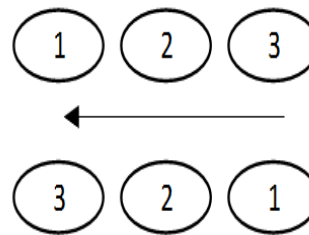

Figure 3. Swap illustration
Figure 4. Flip illustration

3. RESEARCH METHOD

3.1. Model assumptions

Flow shop scheduling problem (FSSP) is a scheduling problem to complete the problem on the m-machine in the same sequence of the n-job. FSSP on this issue considers sequence-dependent setup. Some assumptions in FSSP with dependent setup time are; (1) the sequence of jobs (n=1,2,3, ..., j) done on machines (m=1,2,3, ..., i) is the same, (2) start time for all machines are t=0, (3) the setup time in each job in the machine is dependent sequence job, (4) machine finish is when the last job is finished on each machine, which means each machine that stops independently, (5) setup time is separate from processing time, (7) every job when it starts processing to completion should not be interrupted, and (6) setup time for moving from job j to job k on the machine i is Sijk (where j=k, Sij shows the setup time for j job if the job is the first job in sequence). The objective function in this model aims to minimize the Total Consumption of Energy produced (TCE). Notations in the problem total energy consumption of PFSP employed in this paper is as follows:

j : Machines index, $j = 1,2 \dots, m$
i : Jobs index, i=1,2 …,n
m : total machines
n : total jobs
Si : Setup time of job i in the first sequence on every machine(hour)
Pi,j : processing time of job sequence i on machines j (hour)
$S_{i,i+1}$: Setup time for sequence switching i to $i + 1$ in all machines (hour)
Sej : machine j index of energy consumption setup (watt/hour)
Pej : machine j index of energy consumption processing (watt/hour)
Iej : machine j index of energy consumption when idle (watt/hour)
Tj : finish time of machines j (hour)
Ci,j : finish time of job sequence i at on machines j
Sj : total setup time of machines j (hour)
Bj : total busy time of machines j (hour)
Ij : total idle time of machines j (hour)
TEC : total energy consumption (watt)

In this problem, the objective function in this study is minimizing total energy consumption (TEC). The EES FSSP issue was developed by Li, *et al.* [26]. The EES FSSP model to reduce energy consumption is as follows:

Objective function $Z = \min TEC$ (11)

Subject to :

$$C_{1,1} = S_1 + P_{1,1} \tag{12}$$

$$C_{1,j} = (C_{1,j-1}, S_1) + P_{1,j}, j = 2..m \tag{13}$$

$$C_{i,1} = C_{i-1,1} + S_{i-1,i} + P_{i,1}, i = 2..n \tag{14}$$

$$C_{i,j} = (C_{i,j-1}, S_{i-1,i} + C_{i-1,j}) + P_{i,j}, i = 2..n, j = 2..m \tag{15}$$

$$Bj = \sum_{i=1}^{n} P_{i,j}, \forall j = 1..m \tag{16}$$

$$Sj = \sum_{i=2}^{n} S_{i-1,i} + S_1, \forall j = 1..m \tag{17}$$

$$Tj = \max (C_{i,j}), \forall i = 1..n, j = 1..m \tag{18}$$

$$Ij = Tj - Bj - Sj, \forall j = 1..m \tag{19}$$

$$TEC = \sum_{j=1}^{m} (Bj.Pej + Ij.Iej + Sj.Sej) \tag{20}$$

Formula (11) describes the objective function, which minimizing energy consumption in the FSSP model. The finish time of sequence one on machine 1 formulates in constraint (12). Constraint (13) explains the settlement time of job sequence one the machine 2 to m. The settlement time of sequence i from machine 1 describes in constraint (14). The finish time of job sequence i in machine j is shown in constraint (15). Constraint (16) describes a total processing time machine. The total of setup time is stipulated constraint (17). Constraint (18) explains the completion time of machine j from permutation. Whereas constraint (19) indicates the total idle time of j machine permutation. Finally, constraint (20) describes the permutation TEC (objective function).

3.2. Data and experimental setup

Experimental data from previous studies were used. Some processing time data were based on research conducted by Carlier [27], Heller [28], and Reeves [29]. The setup time for the first order job was generated from uniform random numbers (5,10). The setup time for moving jobs sequence *i* to *i+1* was generated from uniform random numbers (5,10). Uniform random data of consumption of process energy, setup, and idle time series were as follows: uniform (10,20), uniform (5,10), and uniform (1,5). Table 1 portrayed a list of problems from numerical experiments.

Table 1. List of problem

Problem No.	Job	Machine	Instance from
Case 1 13/4	13	4	Carlier [27]
Case 2 10/6	10	6	Carlier [27]
Case 3 100/10	100	10	Heller [28]
Case 4 20/10	20	10	Heller [28]
Case 5 20/15	20	15	Reeves [29]
Case 6 30/10	30	10	Reeves [29]
Case 7 30/15	30	15	Reeves [29]
Case 8 50/10	50	10	Reeves [29]
Case 9 75/20	75	20	Reeves [29]
Case 10 75/20	75	20	Reeves [29]

This research conducted several experiments to determine the hybrid HHO parameters' effect on energy consumption and computing time. Two parameters were used in this experiment include population and iteration. Iteration included three levels, namely iteration 5, 25, and 50. Furthermore, the population consisted of three levels, namely the population 10, 25, and 50. Each data repeatedly experimented for nine times. We used 10 case data jobs and machines as shown in Table 1. Therefore, the experiments were carried out as many as 90 times. Each experiment recorded energy consumption and computational time.

Furthermore, we also tested the effect of parameters on the structure of energy consumption. The structure of energy consumption comprised process energy consumption, setup energy consumption, and idle energy consumption. In this experiment, we implemented case 3, case 8, case 9, and case 10. Furthermore, the best parameters of the results of the experiment were compared with some previous algorithms, including GA [12], NEH [13], CDS [14], ACO [15], and hybrid metaheuristic [10]. The experiment was done by Matlab R16 software on a Windows 10 Intel® Core ™ i5-8250U RAM 4 GB processor. The performance of the Hybrid HHO algorithm was measured by the Efficiency Index (EI). EI formula was presented in (21).

$$\text{EIP} = \frac{proposed\ algorithm}{another\ algorithm} \times 100\% \tag{21}$$

4. RESULTS AND DISCUSSION
4.1. The comparison of various parameters toward energy consumption

The results of the hybrid HHO parameter experiment on energy consumption were shown in Table 2. It reflected that the higher the number of iterations resulted in smaller energy consumption. Furthermore, the lower the number of iterations would lead to greater energy consumption. In population parameters, the higher the population resulted in lower energy consumption. Furthermore, the lower population produced higher energy consumption. Overall these findings were consistent with the findings reported by Tang, *et al.* [30]. It is now generally accepted that population and proper iteration reduce FSSP energy consumption. For the case of small jobs, the best parameter uses population numbers and small iterations. Conversely, for large jobs, the population and iteration used are high.

4.2. The comparison of various parameters toward computation time

Table 2 also displayed the result of the comparison of multiple parameters toward computation time. In iteration parameters, significant iterations increase computational time. Conversely, a small iteration decreases computational time. Furthermore, in the iteration parameter, a large population increases computational time. Conversely, a small population decreases computing time. The number of jobs affects the computational time. Experimental results showed that large jobs increase computational time.

Tabel 2. The results of the experiment parameter variations on energy consumption and computation time

Population	problem	Total Energy Consumption (watt)			Computation Time (second)		
		Iteration			Iteration		
		5	25	50	5	25	50
5	Case 1 13/4	349339	346882	351027	0.033146	0.039977	0.072365
	Case 2 10/6	478592	486000	481260	0.039030	0.039906	0.066424
	Case 3100/10	117640	113443	113253	0.054764	0.168875	0.325313
	Case 4 20/10	22572	22218	21687	0.008835	0.048976	0.101677
	Case 5 20/15	265563	264676	265199	0.032797	0.057065	0.120249
	Case 6 30/10	276188	275944	274736	0.015781	0.058442	0.116066
	Case 7 30/15	425637	425101	422070	0.016743	0.070520	0.150442
	Case 8 50/10	446092	440156	438365	0.017363	0.077573	0.161033
	Case 9 75/20	1200753	1199676	1186720	0.032500	0.165406	0.326697
	Case 10 75/20	1297543	1293532	1291371	0.032946	0.163298	0.345988
25	Case 1 13/4	349545	348105	347698	0.013922	0.079200	0.155404
	Case 2 10/6	483814	481991	479060	0.017884	0.078955	0.159360
	Case 3100/10	117612	116467	108669	0.052846	0.304258	0.577329
	Case 4 20/10	21994	21563	21559	0.019935	0.118536	0.221414
	Case 5 20/15	265674	263438	262466	0.024908	0.130414	0.268450
	Case 6 30/10	275748	275221	272654	0.022985	0.128296	0.250404
	Case 7 30/15	422169	421397	416485	0.026573	0.142162	0.295109
	Case 8 50/10	445739	443778	441927	0.030262	0.165092	0.322063
	Case 9 75/20	1197514	1185167	1184112	0.060237	0.288490	0.633256
	Case 10 75/20	1302907	1287244	1282054	0.056205	0.300946	0.591204
50	Case 1 13/4	347548	347118	347101	0.025392	0.130725	0.299974
	Case 2 10/6	483931	481680	478735	0.027437	0.160287	0.308441
	Case 3100/10	116110	116098	110511	0.091227	0.489995	1110880
	Case 4 20/10	22407	22225	21353	0.037658	0.217429	0.417839
	Case 5 20/15	263945	262178	261877	0.041895	0.255801	0.494134
	Case 6 30/10	275412	273453	273099	0.043486	0.226931	0.477967
	Case 7 30/15	424132	423583	420924	0.048858	0.262267	0.588634
	Case 8 50/10	445371	444536	442709	0.057114	0.308542	0.646978
	Case 9 75/20	1201292	1192656	1189724	0.099352	0.570180	1166938
	Case 10 75/20	1300412	1299754	1296874	0.100612	0.574883	1039619

4.3. The comparison of various parameters toward structure energy consumption

Experimental results of the effect of parameters on structure energy consumption were presented in Table 3. It showed that FSSP energy consumption with the sequence-dependent setup was influenced by setup and idle time. Overall, these findings are consistent with the study results by Utama, *et al*. [10]. Furthermore, the higher iteration and population would decrease the energy consumption setup and idle. Conversely, the smaller iteration and population would increase the energy consumption setup and idle.

Table 3. The parameter influence the structure energy consumption (watt)

Population	problem	Iteration 5			Iteration 25			Iteration 50		
		proses	setup	idle	proses	setup	idle	proses	setup	idle
10	Case 3 100/10	64100	51496	2044	64100	47372	1971	64100	47185	1968
	Case 8 50/10	399421	27180	19491	399421	21415	19320	399421	19716	19228
	Case 9 75/20	1052381	87361	61011	1052381	86311	60984	1052381	73463	60876
	Case 10 75/20	1119568	85016	92959	1119568	81136	92828	1119568	79025	92778
25	Case 3 100/10	64100	51482	2030	64100	50343	2024	64100	42617	1952
	Case 8 50/10	399421	26954	19364	399421	25144	19213	399421	23398	19108
	Case 9 75/20	1052381	84146	60987	1052381	71899	60887	1052381	70942	60789
	Case 10 75/20	1119568	90525	92814	1119568	74928	92748	1119568	69800	92686
50	Case 3 100/10	64100	50004	2006	64100	49985	2013	64100	44491	1920
	Case 8 50/10	399421	26843	19107	399421	26101	19014	399421	24334	18954
	Case 9 75/20	1052381	87947	60964	1052381	79488	60787	1052381	76690	60653
	Case 10 75/20	1119568	88230	92614	1119568	87639	92547	1119568	84854	92452

4.4. The comparison of algorithms

Table 4 indicated the other EIP algorithms against the proposed algorithm. The proposed algorithms were compared with including GA [12], NEH [13], CDS [14], ACO [15], and hybrid metaheuristic [10]. The results suggested that the proposed algorithm was more competitive than other algorithms. This result was proven by the EIP value <100%. The values of EIP including GA [12], NEH [13], CDS [14], ACO [15] and Hybrid Metaheuristic [10] were as follows 98.79%, 97.81%, 97.73%, 98.79%, and 99.00%. Therefore, it was clear that the Hybrid HHO algorithm significantly improves the FSSP energy efficiency solution's quality. Hybrid HHO is an efficient algorithm for solving EES in FSSP with the sequence-dependent setup.

Table 4. EIP algorithm comparison

problem	GA [12]	NEH [13]	CDS [14]	ACO [15]	Hybrid Metaheuristic [10]
Case 1 13/4	99,73%	99,46%	98,11%	99,75%	99,98%
Case 2 10/6	99,99%	98,39%	99,99%	99,54%	99,97%
Case 3100/10	94,69%	92,94%	94,31%	93,53%	94,97%
Case 4 20/10	98,40%	94,29%	93,03%	97,05%	97,64%
Case 5 20/15	99,67%	98,22%	97,95%	99,96%	99,18%
Case 6 30/10	99,51%	98,15%	98,83%	99,37%	99,26%
Case 7 30/15	98,12%	99,06%	98,31%	99,81%	99,82%
Case 8 50/10	98,39%	99,06%	98,99%	99,56%	99,97%
Case 9 75/20	99,54%	98,97%	98,71%	99,49%	99,37%
Case 10 75/20	99,92%	99,52%	99,13%	99,81%	99,80%
Average	98,79%	97,81%	97,73%	98,79%	99,00%

5. CONCLUSION

This study discussed the energy-efficient scheduling (EES) in FSSP with the sequence-dependent setup to minimize energy consumption. This study succeeds in proposing the hybrid Harris hawk optimization (hybrid HHO) algorithm to solve the EES in the FSSP with the sequence-dependent setup problem. The higher the number of iteration and populations can reduce energy consumption. However, these produced significant computation time. Setup and idle time are time structures that influence to reduce energy consumption. Furthermore, the proposed procedure was tested with several algorithms. The test outcome demonstrated that the hybrid HHO algorithm was more effective than other procedures. Further research should be aimed at developing procedures and dynamic job arrival. Future studies also need to develop the hybrid HHO procedure with other metaheuristic procedures to produce more best results.

REFERENCES
[1] X. Wu and A. Che, "A memetic differential evolution algorithm for energy-efficient parallel machine scheduling," *Omega,* vol. 82, pp. 155-165, 2019, doi: 10.1016/j.omega.2018.01.001.
[2] K. Fang, N. Uhan, F. Zhao, and J. W. Sutherland, "A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction," *Journal of Manufacturing Systems,* vol. 30, pp. 234-240, 2011, doi: doi.org/10.1016/j.jmsy.2011.08.004.
[3] S. A. Mansouri, E. Aktas, and U. Besikci, "Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption," *European Journal of Operational Research,* vol. 248, pp. 772-788, 2016, doi: 10.1016/j.ejor.2015.08.064.
[4] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research,* vol. 45, pp. 4247-4271, 2007, doi: 10.1080/00207540701450013.
[5] D. M. Utama, T. Baroto, and D. S. Widodo, "Energy-Efficient Flow Shop Scheduling Using Hybrid Grasshopper Algorithm Optimization," *Jurnal Ilmiah Teknik Industri,* vol. 19, pp. 30-38, 2020, doi: 10.23917/jiti.v19i1.10079.
[6] D. M. Utama, D. S. Widodo, M. F. Ibrahim, K. Hidayat, T. Baroto, and A. Yurifah, "The hybrid whale optimization algorithm: A new metaheuristic algorithm for energy-efficient on flow shop with dependent sequence setup," *Journal of Physics: Conference Series,* vol. 1569, p. 022094, 2020, doi: 10.1088/1742-6596/1569/2/022094.
[7] C. Gahm, F. Denz, M. Dirr, and A. Tuma, "Energy-efficient scheduling in manufacturing companies: A review and research framework," *European Journal of Operational Research,* vol. 248, pp. 744-757, 2016, doi: 10.1016/j.ejor.2015.07.017.
[8] D. M. Utama, D. S. Widodo, M. F. Ibrahim, and S. K. Dewi, "An effective hybrid ant lion algorithm to minimize mean tardiness on permutation flow shop scheduling problem," *International Journal of Advances in Intelligent Informatics,* vol. 6, pp. 23-35, 2020, doi: 10.26555/ijain.v6i1.385.
[9] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of operations research,* vol. 1, pp. 117-129, 1976.
[10] D. Utama, D. Widodo, W. Wicaksono, and L. Ardiansyah, "A new hybrid metaheuristics algorithm for minimizing energy consumption in the flow shop scheduling problem," *International Journal of Technology,* vol. 10, pp. 320-331, 2019, doi: 10.14716/ijtech.v10i2.2194.
[11] D. M. Utama, "An effective hybrid sine cosine algorithm to minimize carbon emission on flow-shop scheduling sequence dependent setup," *Jurnal Teknik Industri,* vol. 20, pp. 62-72, 2019, doi: 10.22219/JTIUMM.Vol20.No1.62-72.
[12] J. Luo, S. Fujimura, D. El Baz, and B. Plazolles, "GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem," *Journal of Parallel and Distributed Computing,* vol. 133, pp. 244-257, 2019, doi: 10.1016/j.jpdc.2018.07.022.
[13] G.-S. Liu, J.-J. Li, and Y.-S. Tang, "Minimizing total idle energy consumption in the permutation flow shop scheduling problem," *Asia-Pacific Journal of Operational Research,* vol. 35, p. 1850041, 2018, doi: 10.1142/S0217595918500410.
[14] D. M. Utama, " Development of NEH and CDS Algorithms to Minimize Energy Consumption in Flow Shop Scheduling (in Bahasa: Pengembangan Algoritma NEH Dan CDS Untuk Meminimasi Consumption Energy Pada Penjadwalan Flow Shop)," in *Prosiding SENTRA (Seminar Teknologi dan Rekayasa)*, pp. 47-54, 2019, doi: 10.22219/sentra.v0i4.2309.
[15] L. Tuo, L. Dai, and X. Chen, "Scheduling of Discrete Manufacturing Process for Energy Saving," in *Applied Mechanics and Materials*, pp. 4248-4254, 2014, doi: 10.4028/www.scientific.net/AMM.556-562.4248.
[16] X. Li, Z. Yang, R. Ruiz, T. Chen, and S. Sui, "An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects," *Information Sciences,* vol. 453, pp. 408-425, 2018, doi: 10.1016/j.ins.2018.04.038.
[17] I. Benkalai, D. Rebaine, C. Gagné, and P. Baptiste, "The migrating birds optimization metaheuristic for the permutation flow shop with sequence dependent setup times," *IFAC-PapersOnLine,* vol. 49, pp. 408-413, 2016, doi: 10.1016/j.ifacol.2016.07.640.
[18] R. Ramezanian, M. M. Vali-Siar, and M. Jalalian, "Green permutation flowshop scheduling problem with sequence-dependent setup times: a case study," *International Journal of Production Research,* vol. 57, pp. 3311-3333, 2019, doi: 10.1080/00207543.2019.1581955.
[19] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems,* vol. 97, pp. 849-872, 2019, doi: 10.1016/j.future.2019.02.028.
[20] H. Chen, S. Jiao, M. Wang, A. A. Heidari, and X. Zhao, "Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts," *Journal of Cleaner Production,* vol. 244, p. 118778, 2020, doi: 10.1016/j.jclepro.2019.118778.

[21] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Parameters extraction of three-diode photovoltaic model using computation and Harris Hawks optimization," *Energy,* vol. 195, p. 117040, 2020, doi: 10.1016/j.energy.2020.117040.

[22] A. R. Yıldız, B. S. Yıldız, S. M. Sait, S. Bureerat, and N. Pholdee, "A new hybrid harris hawks-Nelder-Mead optimization algorithm for solving design and manufacturing problems," *Materials Testing,* vol. 8, no. 61, pp. 735-743, 2019, doi: 10.3139/120.111378.

[23] A. R. Yıldız, B. S. Yıldız, S. M. Sait, and X. Li, "The Harris hawks, grasshopper and multi-verse optimization algorithms for the selection of optimal machining parameters in manufacturing operations," *Materials Testing,* vol. 8, no. 61, pp. 1-15, 2019, doi: 10.3139/120.111377.

[24] X. Bao, H. Jia and C. Lang, "A Novel Hybrid Harris Hawks Optimization for Color Image Multilevel Thresholding Segmentation," in *IEEE Access*, vol. 7, pp. 76529-76546, 2019, doi: 10.1109/ACCESS.2019.2921545.

[25] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software,* vol. 55, pp. 10-31, 2013, doi: 10.1016/j.advengsoft.2012.09.003.

[26] S. Li, F. Liu, and X. Zhou, "Multi-objective energy-saving scheduling for a permutation flow line," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 232, pp. 879-888, 2018, doi: 10.1177/0954405416657583.

[27] J. Carlier, "Ordonnancements à contraintes disjonctives," *RAIRO-Operations Research,* vol. 12, pp. 333-350, 1978.

[28] J. Heller, "Some Numerical Experiments for an M × J Flow Shop and its Decision-Theoretical Aspects," *Operations Research,* vol. 8, pp. 178-184, 1960.

[29] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Computers & Operations Research,* vol. 22, pp. 5-13, 1995, doi: 10.1016/0305-0548(93)E0014-K.

[30] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in Industry,* vol. 81, pp. 82-95, 2016, doi: 10.1016/j.compind.2015.10.001.

BIOGRAPHIES OF AUTHORS

Dana Marsetiya Utama is currently a Lecturer and a Researcher in the Industrial Engineering Department, University of Muhammadiyah Malang indonesia. His research interests include optimization engineering, scheduling, and modeling. He received the bachelor's degree in industrial engineering from Trunojoyo University, in 2008, and the master's degree in industrial engineering from the University of brawijaya, in 2011.

Dian Setiya Widodo is currently a Lecturer and a Researcher in the department of Manufacturing Technology, Vocational Faculty, University of 17 Agustus 1945 Surabaya Indonesia. His research interests include optimization engineering, scheduling, and modeling. He received the bachelor's degree in industrial engineering from Trunojoyo University, in 2011, and the master's degree in industrial engineering from the University of brawijaya, in 2014.