An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup

Dana Marsetiya Utama
Department Industrial Engineering, Universitas Muhammadiyah Malang, Indonesia
Jalan Raya Tlogomas No. 246, Malang 65144, Indonesia
Corresponding author: dana@umm.ac.id

ARTICLE INFO

Article history
Received August 19, 2018
Revised January 5, 2018
Accepted February 20, 2019
Available Online February 28, 2019

Keywords
Emissions
Sine Cosine
flow shop
Scheduling

ABSTRACT

Recently, carbon emissions have become a major environmental problem. In the industrial sector, carbon emissions account for half of the world's total carbon emissions. This article discusses the issue of scheduling Flow Shop Sequence Dependent Setup (FSSDS). It aims to minimize carbon emissions. The algorithm proposed is the Hybrid Sine Cosine Algorithm (HSCA) to solve FSSDS problems to reduce carbon emissions. We offered one of some search agents in the SCA using NEH. The algorithm is used for some tests on different jobs and machines. Several experiments were carried out to test the parameters and effectiveness of the algorithm. The parameters used in the trial are population and iteration. As a result, several parameters were proposed to HSCA to minimize carbon emissions. In the effectiveness test, the HSCA showed better performance compared to the simulated annealing and cross-entropy algorithm.

1. Introduction

Recently, global warming and increasing carbon emissions have become major environmental problems. The industrial sector accounts for half of the world's total carbon emissions [1]. The energy consumption of this sector has almost doubled over the past 60 years [2]. The increase in carbon emissions raises concerns related to climate change. Manufacturing companies face pressure to reduce carbon emissions. Environmental and economic factors motivate to reduce energy consumption and emissions from manufacturing companies [3-5]. At present, the problem of minimizing energy consumption and emissions is the focus of researchers. One effective way is scheduling. Proper scheduling can minimize carbon emissions. Scheduling is the arrangement of resources to carry out many jobs at a particular time [6, 7]. Generally, some jobs are produced in the same process sequence [8]. These problems are categorized as Flow Shop

https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72 http://ejournal.umm.ac.id/index.php/industri ti.jurnal@umm.ac.id

Please cite this article as: Utama, Dana Marsetiya. (2019). An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup. Jurnal Teknik Industri, 20(1), 62-72. doi:https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72

Scheduling Problem (FSSP). Generally, FSSP aims to complete jobs to minimize completion time [8]. Flow Shop Sequence Dependent Setup (FSSDS) is an FSSP development problem. In FSSDS, the machine setup time is dependent on the machine sequence. In this problem, the order of jobs affects the setup time [9].

In the flow shop problem, some procedures are examined by researchers. Heuristic algorithms that are popularly used in the case of flow shops are Campble Dudek Smith (CDS) [10], Branch and bound [11], and Nawaz Encore Ham (NEH) [12]. Furthermore, several metaheuristic algorithms that are often used are simulated annealing [13, 14] and the immune system algorithm [15]. Some researchers have also examined the problems of flow shop blocking [16], flow shop with maintenance activities [17]. Furthermore, Riahi, et al. [18] examined the flow shop problem with search spread. Karimi and Davoudpour [19] proposed a competitive colonial algorithm for the multi-objective Hybrid flow shop problem. Hatami, et al. [20] used the colonial competitive algorithm for the flow shop problem with the initial setup time. Generally, researchers use the minimization completion time performance.

At present, research minimizing energy and carbon emissions has not been much studied. However, several researchers have studied this problem. Utama [21]proposed the CDS and NEH algorithm to minimize energy consumption. Liu, et al. [22] proposed a hybrid fruit fly algorithm for solving flexible job-shop to maximize carbon footprint. Zeng, et al. [23] proposed the NSGA II algorithm to minimize energy consumption. Tang, et al. [24] proposed the PSO algorithm to minimize carbon emissions. Piroozfard, et al. [25] proposed minimizing carbon emissions using the Genetic algorithm. Li, et al. [26] used a hybrid optimization approach to minimize carbon emissions. Wu and Che [27] proposed minimizing carbon emissions in parallel machines using the Memetic Differential Evolution algorithm. Batista Abikarram, et al. [28] conducted minimizing carbon emissions in parallel machines. Tan, et al. [29] proposed scheduling to minimize carbon emissions with MILP.

The above review shows that several studies have been conducted. However, to our knowledge, no researchers have used Sine Cosine Algorithm (SCA) to minimize carbon emissions. Therefore, this study aims to develop a new algorithm to solve the FSSDS problem. We propose the Sine Cosine Algorithm (HSCA) hybrid algorithm to solve this problem. The proposed algorithm is used to minimize carbon emissions in the FSSDS problem. The Sine-cosine (SCA) algorithm is an optimization algorithm proposed by Mirjalili [30]. The SCA algorithm refers to the No Free Lunch (NFL) theorem. The NFL theorem allows researchers to propose new algorithms or improve current procedures. This algorithm can be used to solve problems in different fields. This SCA algorithm is based on the cosine sine of mathematical functions to solve optimization problems.

2. Methods

2.1 Nomenclature and Problem Definition

The problem of scheduling the Flow Shop Sequence Dependent Setup (FSSDS) has a t_{ij} processing time, and n jobs ($n = 1,2,3,..,i$) that are processed on the m machine ($m = 1,2,3,..j$). Some FSSDS assumptions are (1) the sequence of n jobs ($n = 1,2,3,...,j$) done on each m of machines ($m = 1,2,3,...,i$) is the same. (2) the processing time for each job is Pij. Pij shows the time of the $i - job$ and is done on the $j - machine$. (3) All machines are available when $t = 0$. (4) The setup time depends on the work order, and it is separated from the processing time. (5) the time set up job i to job $i + 1$ on machine j is $S_{i,i+1}$. Furthermore, S_i shows the setup time for job i if the job i is the first order job. (6) Each job, when it starts processing to completion, must be in order and must not be interrupted.

Jurnal Teknik Industri
Vol. 20, No. 1, February 2019, pp. 62-72

(7) Each machine starts at time = 0 and finishes when the last job on each machine is finished (each machine that stops independently of other machines). The notation used in scheduling carbon emissions minimization is as follows:

i	:	index of job, $i = 1,2 \ldots, n$
j	:	index of Machine, $j = 1,2 \ldots, m$
n	:	total jobs
m	:	total Machine
S_i	:	set up time Job i on sequence 1 (kilowatt)
$S_{i,i+1}$	:	Set-up time for job i and next job sequence $i+1$ (hours)
ST_i	:	total setup time (hours)
Pi,j	:	Job processing time sequence i on machine j (hours)
Pej	:	processing energy consumption on Machine j (kilowatt)
Sej	:	Energy consumption setup on Machine j (kilowatt)
Iej	:	energy consumption idle to Machine j (kilowatt)
Ci,j	:	time of completion of sequence i job on machine j
Tj	:	time of completion on the machine j
Bj	:	total busy time on machine j
Ij	:	total idle time on the Machine j
CEj	:	carbon emissions emitted by Machine j (kg/kilowatt)
TCE	:	total carbon emissions (kg)

The purpose of this model is to minimize carbon emissions (TCE) [31]. Following is the FSSDS formula for minimizing the total carbon emissions:

Minimize TCE (1)

Subject to

$$C_{1,1} = P_{1,1} + S_1 \tag{2}$$
$$C_{1,j} = \max(C_{1,j-1}, S_1) + P_{1,j}, \quad j = 2..m \tag{3}$$
$$C_{i,1} = C_{i-1,1} + S_{i,i+1} + P_{i,1}, \quad i = 2..n \tag{4}$$
$$C_{i,j} = \max(C_{i,j-1}, S_{i,i+1} + C_{i-1,j}) + P_{i,j}, \; i = 2..n, \quad j = 2..m \tag{5}$$
$$Bj = \sum_{j=1}^{n} Pij \tag{6}$$
$$STj = Si + \sum_{i=2}^{m} S_{i,i+1}, \quad j = 1..m \tag{7}$$
$$Ij = Tj - (Bj + STj), \quad j = 1..m \tag{8}$$
$$TCE = \sum_{j=1}^{n}(Bj.Pej.CEj + STj.Sej.CEj + Ij.Iej.CEj) \tag{9}$$

Equation (1) formulates the objective of minimizing carbon emissions. Equation (2) illustrates the formula for completing job sequence 1 in machine 1. Equation (3) shows the completion time of job sequence one on machines 2 to j. Equation (4) describes the completion time of a sequence i job on machine 1. Equation (5) illustrates the completion time of a sequence i job machine j. Equation (6) explains the machines busy time j. Equation (7) illustrates the total setup time on machine j. Equation (8) shows the total idle time on machine j. Equation (9) illustrates the formula for calculating total carbon emissions.

2.2 Hybrid Sine Cosine Algorithm (HSCA) algorithm proposed

We proposed Hybrid Sine Cosine Algorithm (HSCA) for minimizing total carbon emissions in the FSSDS problem. The HSCA algorithm also has the characteristics to solve scheduling. We called this algorithm the Hybrid Sine Cosine Algorithm (HSCA). We added the NEH algorithm to improve HSCA performance. In this article, the NEH algorithm replaced one search agent in the SCA. There are four steps in the HSCA algorithm: initializing positions for search agents, applying LRV for sequence, changing one search agent position using the NEH algorithm, and evaluating SCA. Pseudo-code proposed algorithm is presented in Algorithm 1.

Algorithm 1 Pseudo-code Sine-cosine algorithm

1. Search agent initialization (solution) (X)
2. Apply LRV on each search agent to be mapped into job permutation
3. Solve the PFSSP using NEH
4. Choose about one search agents from the population and replace them with NEH
5. **For** i=1: maximum number of iterations
6. Evaluate each search agent with an objective function
7. Update the best solution obtained from (P = X *)
8. Update r1, r2, r3 and r4
9. Update the search agent position using equation (12)
10. **End for**
11. The best global return is obtained so far as the optimum

2.2.1 Nawaz Enscore Ham (NEH) Algorithm

Nawaz Enscore Ham (NEH) is a practical heuristic algorithm for FSSP cases. The workings of the NEH method are jobs with the most significant total processing time given top priority to be carried out [12]. Initialize NEH job sequences descending based on the total processing time of each job. Furthermore, each job position's best order is determined (see Algorithm 2) [32].

Algorithm 2 Pseudo-code Nawaz Enscore Ham (NEH) algorithm

1. Compute the total processing time for each job on m machine
2. Generate a sequence j = (j_1, j_2,…, j_n) by sorting the jobs in non-increasing order according to the total processing time
3. The first job is taken. $\pi_* = \{ j_1 \}$
4. **for** $i = 1 : n - 1$
5. Take job j_i form j and insert j_i into all possible positions of π_*;
6. Evaluate the new sequence $\pi \leftarrow \pi_* \cup j_i$ (use equations 1 to 9);
7. Select the $\pi_* \leftarrow \pi$ with the lowest objective value;
8. **End for**
9. return π_*;

2.2.2 Initialize Search Agent position

The initial position of the search agent is generated randomly. Agent search position based on upper bound and lower bound. It must be ensured that there is no similar loop in the initialization of the search agent position. Fig. 1 illustrates the position of the search agent. The search agent position is a matrix.

2.2.3 Apply Large Rank Value (LRV)

We proposed a ranking order for search agents using Large Rank Value (LRV). LRV procedure is sorting from the largest to the smallest value. Fig. 2 illustrates an illustration of how the LRV works.

Jurnal Teknik Industri
Vol. 20, No. 1, February 2019, pp. 62-72

$$\begin{bmatrix} 0{,}71 & 0{,}58 & 0{,}62 \\ 0{,}39 & 0{,}95 & 0{,}46 \end{bmatrix} \quad \begin{bmatrix} 0{,}71 & 0{,}58 & 0{,}58 \\ 0{,}39 & 0{,}95 & 0{,}46 \end{bmatrix}$$

Correct Position Wrong Position

Fig. 1. Illustrates Initialization

0,58	0,32	0,86

↓

J1	J2	J3
2	3	1

Fig. 2. Illustrates LRV

2.2.4 Sine-Cosine Algorithm (SCA)

We proposed a Hybrid Sine-cosine algorithm that has been proposed by Mirjalili [28]. The basic principle of the SCA algorithm is to combine random solutions with a high degree of randomness to find the optimal solution from the search space. The Sine and Cosine effects on formulated in Equations (10) and (11).

$$X_i^{t+1} = X_i^t + r_1.sin(r_2).|r_3P_i^t - X_i^t| \tag{10}$$

$$X_i^{t+1} = X_i^t + r_1.\cos(r_2).| r_3P_i^t - X_{ij}^t | \tag{11}$$

X_i^t shows the position of the solution during the $i-th$ dimension in the $t-iteration$. $r1$ / $r2$ / $r3$ describe random numbers. P_i is the position of the destination point in the $i-th$ dimension. Moreover, | | show absolute value. These two equations are combined as in equation (12).

$$X_i^{t+1} = \begin{cases} X_i^t + r_1.\sin(r_2).|r_3P_i^t - X_i^t|, r_4 < 0.5 \\ X_i^t + r_1.\cos(r_2).|r_3P_i^t - X_i^t|, r_4 \geq 0.5 \end{cases} \tag{12}$$

Equation (12) shows that there are four main parameters in SCA. These are $r1$, $r2$, $r3$, and $r4$. The parameter $r1$ determines the next position area in the form of space between the solution and the destination. The parameter $r2$ determines how far the movement must go in or out of the destination. The $r3$ parameter is a random weight. Parameter $r4$ shows the random number of transitions between the sine and cosine components in Equation (12). Random location is achieved by defining the number $r2$ in the range [0, 2п] in equation (12). This algorithm can balance exploration and exploitation to find the optimal global solution. To balance the exploration and exploitation phase, sines and cosines in Equations (10) through (12) are changed adaptively using equation (13). t is the current iteration. T indicates the maximum number of iterations. Moreover, a is a constant. General steps of the SCA Algorithm are presented in Algorithm 3.

$$r_1 = a - t\frac{a}{T} \tag{13}$$

Algorithm 3 Pseudo-code algorithm Algoritma Sine-cosine

1. Search agent initialization (solution) (X)
2. For i=1: maximum number of iterations
3. Evaluate each search agent with an objective function
4. Update the best solution obtained from (P = X *)
5. Update r1, r2, r3 and r4
6. Update the search agent position using equation (12)
7. **End for**
8. The best global return is obtained so far as the optimum

Please cite this article as: Utama, Dana Marsetiya. (2019). An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup. Jurnal Teknik Industri, 20(1), 62-72. doi:https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72

 ISSN : 1978-1431 print | 2527-4112 online

2.3 Experiment Procedures

This study used seven variations of the job. Furthermore, this study used two various machines. The total experimental environment in this study was 14. Generation of processing time, job set-up time in the first sequence, and set-up time for sequences from job i to job $i + 1$ were uniform (10.50), uniform (1, 10), and uniform (1,10). The value of the generation of uniform distributions Process energy consumption, Set up energy consumption, and idle engine energy consumption was (5.10), (1.2), and (1.3), respectively. Carbon emissions from each engine were obtained with uniform parameter values (0.1).

In the parameter experiment, the parameters used the experiments are population and iteration. The HSCA population factor consisted of 3 levels: population 10, 50, and 100. Iteration factor consisted of 5 levels as 10, 50, 100, 200, and 500. For every data, there were 15 experiments performed. The study had 14 different experimental environments. Thus, the experiment was carried out as much as 210 times the treatment. To test the effectiveness of the algorithm, the study conducted a comparison with several algorithms. The algorithm chosen for comparison was simulated annealing and genetic algorithm. The algorithm was a popular algorithm at the moment. HSCA parameters used are 100 populations and 500 iterations. Comparative testing used 100 job problems and 16 machines. The problem was repeated 20 times. The comparative test used an independent sample t-test. Numerical tests were carried out in the Matlab R16 software on a Windows 8.1 AMD 4 x86-64 4 GB processor. A variety of population, job, machine, and iterations are used to find the most optimal possibilities for minimizing carbon emotions.

3. Result and Discussions

This section explains the effectiveness of the proposed algorithm, the results of total carbon emissions, and the computational time of each experiment. We reported the results of developing the SCA algorithm to solve optimization problems. The results of experiments conducted on the Matlab application produce total carbon emissions and computational time according to population, number of jobs, number of machines, and number of iterations.

3.1 Results HSCA parameters toward carbon emissions

Table 1 is a recapitulation of the results of the HSCA experiment on carbon emissions. Based on the results of the experiment, there are several findings from the results of the experiment. The higher the job and when the number of machines is the same, the more significant carbon emissions are produced. The higher the number of machines and when the number of jobs is the same, the carbon emissions produced will be even higher. The higher the population used, the smaller the carbon emissions produced. Thus, carbon emissions are influenced by population. If the iteration is used, the more significant the carbon emission value will be smaller. In small jobs, the parameters that should be used are small populations and small iterations. Furthermore, in large jobs, the parameters that should be used are large populations and significant iterations.

3.2 Testing HSCA parameters for computational time

Table 2 shows the recapitulation of computational time for various possible populations, jobs, machines, and the number of iterations. The findings of several

Jurnal Teknik Industri
Vol. 20, No. 1, February 2019, pp. 62-72

experiments are as follows. The higher the job used, the greater the computational time required. The higher the number of machines used, the greater the required computing time. The higher the population, the greater the computational time. Thus, computing time is influenced by the population. If the iteration gets high, then the computing time also gets high. Thus, computing time is affected by the number of iterations.

Table 1. Results of experiments on carbon emissions (kg)

Population	Job	Machine	Iterations				
			10	50	100	200	500
10	5	4	397.2672	397.2672	397.2672	397.2672	397.2672
	5	16	12327	12283	12283	12283	12283
	10	4	6429.8	6398.3	6403.2	6398.5	6394.5
	10	16	25684	25570	25406	25328	25442
	20	4	5592.5	5592.6	5588.3	5588.3	5587.8
	20	16	41821	41527	41397	41449	41425
	40	4	17476	17471	17468	17453	17455
	40	16	99262	99223	99100	99050	98976
	60	4	33861	33826	33770	33814	33796
	60	16	109200	10911	108940	108770	108840
	80	4	57749	57776	57692	57657	57690
	80	16	189640	18964	189500	189310	189270
	100	4	63490	63450	63466	63440	63446
	100	16	189070	18907	188590	188660	188450
50	5	4	397.267	397.267	397.267	397.2672	397.267
	5	16	12283	12283	12283	12283	12283
	10	4	6410.9	6403.2	6395.6	6394.5	6394.5
	10	16	25497	25518	25389	25311	25347
	20	4	5588.5	5589.2	5587.6	5580.1	5582.8
	20	16	41498	41406	41404	41378	41303
	40	4	17471	17468	17459	17459	17449
	40	16	99231	99075	99008	98931	98871
	60	4	33844	33814	33801	33776	33759
	60	16	108950	108910	108730	108860	108550
	80	4	57758	57692	57675	57678	57663
	80	16	189570	189290	189150	189270	189290
	100	4	63469	63470	63446	63427	63425
	100	16	188800	188750	188570	188710	188230
100	5	4	397.267	397.267	397.267	397.267	397.267
	5	16	12283	12283	12283	12283	12283
	10	4	6396	6392.4	6392.1	6394.5	6391.2
	10	16	25466	25329	25423	25393	25370
	20	4	5583.9	5585.2	5582.6	5584.8	5581.4
	20	16	41641	41423	41359	41350	41267
	40	4	17465	17456	17460	17456	17454
	40	16	99005	98884	98949	98913	98872
	60	4	33782	33736	33773	33770	33736
	60	16	108920	108790	108810	108640	108670
	80	4	57721	57692	57676	57640	57651
	80	16	189610	189340	189320	189150	188910
	100	4	63446	63449	63418	63415	63432
	100	16	188900	188580	188340	188350	188420

Please cite this article as: Utama, Dana Marsetiya. (2019). An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup. Jurnal Teknik Industri, 20(1), 62-72. doi:https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72

Table 2. The results of experiments on computational time (second)

Population	Job	Machine	Iteration				
			10	50	100	200	500
10	5	4	0.125	0.3906	0.375	0.6563	1.4219
	5	16	0.0938	0.2188	0.4063	0.7031	1.5469
	10	4	0.0625	0.25	0.3438	0.7188	1.3906
	10	16	0.0938	0.2031	0.3906	0.6875	1.6094
	20	4	0.125	0.3438	0.6719	0.625	1.6563
	20	16	0.0938	0.2813	0.4063	0.7188	1.8594
	40	4	0.0469	0.2031	0.3594	0.6094	1.5781
	40	16	0.0938	0.2969	0.4844	0.875	2.1406
	60	4	0.0938	0.1875	0.4531	0.7031	1.5
	60	16	0.1406	0.3281	0.5156	1.0313	2.2969
	80	4	0.0781	0.2656	0.4844	0.7656	2.0469
	80	16	0.125	0.3281	0.6563	1.4844	3.1406
	100	4	0.125	0.4063	0.5313	0.7656	2
	100	16	0.1563	0.3906	0.6875	1.3594	3.0313
50	5	4	0.2813	0.7344	1.1406	2.5313	5.3906
	5	16	0.2188	0.7188	1.3438	2.5781	6.3438
	10	4	0.1875	0.5781	1.0938	2.375	5.5156
	10	16	0.1875	0.6875	1.3594	2.7031	6.7656
	20	4	0.3281	0.75	1.2813	2.4531	5.875
	20	16	0.2031	0.9531	1.75	3.0938	7.8438
	40	4	0.2031	0.75	1.2969	2.5313	6.1563
	40	16	0.2969	0.9375	1.8906	3.6719	9.0625
	60	4	0.2344	0.7969	1.4219	2.875	6.6875
	60	16	0.3125	1.1406	2.0938	4.9688	12.4063
	80	4	0.2813	0.7969	1.8281	3.6875	8.3906
	80	16	0.3594	1.4531	3.1875	6.3125	14.75
	100	4	0.2813	0.9375	1.6094	3.1563	7.5781
	100	16	0.4063	1.4688	2.8281	5.4219	13.2031
100	5	4	0.2969	1.0781	2.375	4.1719	10.2656
	5	16	0.4219	1.2656	2.4844	5.0156	12.25
	10	4	0.5	1.2188	2.1875	4.5625	10.5781
	10	16	0.3125	1.3281	2.6875	5.2813	13.3438
	20	4	0.2969	1.1563	2.3281	4.5156	11.1875
	20	16	0.4063	1.6875	3.2344	6.6406	15.1094
	40	4	0.4063	1.3125	2.5781	4.9688	12.0781
	40	16	0.5469	1.9375	3.5313	7.2344	17.6094
	60	4	0.3594	1.4063	2.8906	5.3125	13.0781
	60	16	0.5625	2.6875	4.6719	9.4531	22.6563
	80	4	0.4219	2.0781	3.2656	6.6875	16.4688
	80	16	0.6094	2.6875	6.5469	11.0938	28.6719
	100	4	0.4063	1.6563	3.0469	5.9063	14.8906
	100	16	0.75	2.9219	5.5625	10.4063	26.0938

Please cite this article as: Utama, Dana Marsetiya. (2019). An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup. Jurnal Teknik Industri, 20(1), 62-72. doi:https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72

3.3 HSCA Testing of Other Algorithms

Based on the independent sample t-test (Table 3 and Table 4), the SCA algorithm performs better than the CE and SA algorithms. The average value of carbon SCA emissions produced in the experiment is 200021.50 kg. Furthermore, the CE and SA algorithms produce carbon emissions of 206541.65 kg and 205451.32 kg. The independent t-test statistic test shows ALO algorithm has a significant difference from the CE and SA algorithms.

Table 3. Descriptive summary of the algorithm comparison

Algorithm	Number Experimental	Average Carbon emission (kg)	Std. Deviation (Kg)	Std. Error Mean (Kg)
SCA	20	200021.50	1774.21	324.62
CE	20	206541.65	4213.28	914.21
SA	20	205451.32	3541.20	782.88

Table 4. Independent t-test results

Algorithm	t-value	Sig. (2-tailed)	Decision
SCA-CE	-5.756	0.000	Significant of differences
SCA-SA	-7.216	0.000	Significant of differences

4. Conclusion

The results of the research show that the higher the amount of work, the greater the carbon emissions produced. If the work is significant, the optimal solution is to use a large population and iteration. Moreover, conversely, if the amount of work is small, the optimal solution is to use a small population and iteration. The experimental results show that the HSCA algorithm helps solve FSSDS problems. Suggestions for further research are to improve the algorithm by providing new, more active steps to reduce computational time.

References

[1] S. Shafiee and E. Topal, "When will fossil fuel reserves be diminished?," *Energy Policy,* vol. 37, pp. 181-189, 2009. https://doi.org/10.1016/j.enpol.2008.08.016.

[2] K. Aleklett, M. Höök, K. Jakobsson, M. Lardelli, S. Snowden, and B. Söderbergh, "The Peak of the Oil Age – Analyzing the world oil production Reference Scenario in World Energy Outlook 2008," *Energy Policy,* vol. 38, pp. 1398-1414, 2010. https://doi.org/10.1016/j.enpol.2009.11.021.

[3] T. Simunic, L. Benini, and G. D. Micheli, "Energy-efficient design of battery-powered embedded systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 9, pp. 15-28, 2001. https://doi.org/10.1109/92.920814.

[4] V. Swaminathan and K. Chakrabarty, "Energy-conscious, deterministic I/O device scheduling in hard real-time systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 22, pp. 847-858, 2003. https://doi.org/10.1109/TCAD.2003.814245.

[5] T. Gutowski, C. Murphy, D. Allen, D. Bauer, B. Bras, T. Piwonka, *et al.*, "Environmentally benign manufacturing: Observations from Japan, Europe and

the United States," *Journal of Cleaner Production,* vol. 13, pp. 1-17, 2005. https://doi.org/10.1016/j.jclepro.2003.10.004.

[6] S. Harto, A. K. Garside, and D. M. Utama, "Penjadwalan Produksi Menggunakan Algoritma Jadwal Non Delay Untuk Meminimalkan Makespan (studi kasus di CV. Bima Mebel)," *Spektrum Industri,* vol. 14, pp. 79-88, 2015. http://journal.uad.ac.id/index.php/Spektrum/article/view/3706.

[7] D. M. Utama, "Analisa Perbandingan Penggunaan Aturan Prioritas Penjadwalan Pada Penjadwalan Non Delay N Job 5 Machine," *Prosiding SENTRA (Seminar Teknologi dan Rekayasa),* vol. 2, pp. 19-23, 2017. http://research-report.umm.ac.id/index.php/research-report/article/view/1180.

[8] I. Masudin, D. M. Utama, and F. Susastro, "Penjadwalan Flowshop Menggunakan Algoritma Nawaz Enscore Ham," *Jurnal Ilmiah Teknik Industri,* vol. 13, pp. 54-59, 2014. https://doi.org/10.23917/jiti.v13i1.321.

[9] 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. https://doi.org/10.1016/j.ifacol.2016.07.640.

[10] H. G. Campbell, R. A. Dudek, and M. L. Smith, "A heuristic algorithm for the n job, m machine sequencing problem," *Management science,* vol. 16, pp. B-630-B-637, 1970. https://doi.org/10.1287/mnsc.16.10.B630.

[11] D. M. Utama, "Algoritma LPT-Branch and Bound Pada Penjadwalan Flexible Flowshop untuk Meminimasi Makespan," *PROZIMA (Productivity, Optimization and Manufacturing System Engineering),* vol. 2, pp. 20-26, 2018. http://ojs.umsida.ac.id/index.php/prozima/article/view/1527.

[12] M. Nawaz, E. E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega,* vol. 11, pp. 91-95, 1983. https://doi.org/10.1016/0305-0483(83)90088-9.

[13] M. Husen, I. Masudin, and D. Utama, "Penjadwalan job shop statik dengan metode simulated annealing untuk meminimasi waktu makespan," *Spektrum Industri,* vol. 13, pp. 115-228, 2015. http://dx.doi.org/10.12928/si.v13i2.2689.

[14] M. Firdaus, I. Masudin, and D. M. Utama, "Penjadwalan Flowshop Dengan Menggunakan Simulated Annealing," *Spektrum Industri,* vol. 13, pp. 27-40, 2015. http://dx.doi.org/10.12928/si.v13i1.1836.

[15] R. Nasution, A. K. Garside, and D. M. Utama, "Penjadwalan Job Shop Dengan Pendekatan Algoritma Artificial Immune System," *Jurnal Teknik Industri,* vol. 18, pp. 29-42, 2017. https://doi.org/10.22219/JTIUMM.Vol18.No1.29-42.

[16] V. Riahi, M. A. H. Newton, K. Su, and A. Sattar, "Constraint guided accelerated search for mixed blocking permutation flowshop scheduling," *Computers & Operations Research,* vol. 102, pp. 102-120, 2019. https://doi.org/10.1016/j.cor.2018.10.003.

[17] C.-Y. Cheng, K.-C. Ying, H.-H. Chen, and J.-X. Lin, "Optimization algorithms for proportionate flowshop scheduling problems with variable maintenance activities," *Computers & Industrial Engineering,* vol. 117, pp. 164-170, 2018. https://doi.org/10.1016/j.cie.2018.01.027.

[18] V. Riahi, M. Khorramizadeh, M. A. Hakim Newton, and A. Sattar, "Scatter search for mixed blocking flowshop scheduling," *Expert Systems with Applications,* vol. 79, pp. 20-32, 2017. https://doi.org/10.1016/j.eswa.2017.02.027.

[19] N. Karimi and H. Davoudpour, "Multi-objective colonial competitive algorithm for hybrid flowshop problem," *Applied Soft Computing,* vol. 49, pp. 725-733, 2016. https://doi.org/10.1016/j.asoc.2016.06.034.

[20] S. Hatami, L. Calvet, V. Fernández-Viagas, J. M. Framiñán, and A. A. Juan, "A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem," *Simulation Modelling Practice and Theory,* vol. 86, pp. 55-71, 2018. https://doi.org/10.1016/j.simpat.2018.04.005.

[21] D. M. Utama, "Pengembangan Algoritma NEH Dan CDS Untuk Meminimasi Consumption Energy Pada Penjadwalan Flow Shop," *Prosiding SENTRA (Seminar Teknologi dan Rekayasa),* vol. 4, pp. 47-54, 2018. http://research-report.umm.ac.id/index.php/sentra/article/view/2309.

[22] Q. Liu, M. Zhan, F. O. Chekem, X. Shao, B. Ying, and J. W. Sutherland, "A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint," *Journal of Cleaner Production,* vol. 168, pp. 668-678, 2017. https://doi.org/10.1016/j.jclepro.2017.09.037.

[23] Z. Zeng, M. Hong, Y. Man, J. Li, Y. Zhang, and H. Liu, "Multi-object optimization of flexible flow shop scheduling with batch process — Consideration total electricity consumption and material wastage," *Journal of Cleaner Production,* vol. 183, pp. 925-939, 2018. https://doi.org/10.1016/j.jclepro.2018.02.224.

[24] 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. https://doi.org/10.1016/j.compind.2015.10.001.

[25] H. Piroozfard, K. Y. Wong, and W. P. Wong, "Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm," *Resources, Conservation and Recycling,* vol. 128, pp. 267-283, 2018. https://doi.org/10.1016/j.resconrec.2016.12.001.

[26] X. Li, W. Li, X. Cai, and F. He, "A hybrid optimization approach for sustainable process planning and scheduling," *Integrated Computer-Aided Engineering,* vol. 22, pp. 311-326, 2015. https://doi.org/10.3233/ICA-150492

[27] X. Wu and A. Che, "A memetic differential evolution algorithm for energy-efficient parallel machine scheduling," *Omega,* vol. 82, pp. 155-165, 2019. https://doi.org/10.1016/j.omega.2018.01.001.

[28] J. Batista Abikarram, K. McConky, and R. Proano, "Energy cost minimization for unrelated parallel machine scheduling under real time and demand charge pricing," *Journal of Cleaner Production,* vol. 208, pp. 232-242, 2019. https://doi.org/10.1016/j.jclepro.2018.10.048.

[29] M. Tan, Y. Chen, Y.-x. Su, S.-h. Li, and H. Li, "Integrated optimization model for industrial self-generation and load scheduling with tradable carbon emission permits," *Journal of Cleaner Production,* vol. 210, pp. 1289-1300, 2019. https://doi.org/10.1016/j.jclepro.2018.11.005.

[30] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems,* vol. 96, pp. 120-133, 2016. https://doi.org/10.1016/j.knosys.2015.12.022.

[31] G. I. Galinato and J. K. Yoder, "An integrated tax-subsidy policy for carbon emission reduction," *Resource and Energy Economics,* vol. 32, pp. 310-326, 2010. https://doi.org/10.1016/j.reseneeco.2009.10.001.

[32] Y. B. Park, C. D. Pegden, and E. E. Enscore, "A survey and evaluation of static flowshop scheduling heuristics," *International Journal of Production Research,* vol. 22, pp. 127-141, 1984. https://doi.org/10.1080/00207548408942436.

Please cite this article as: Utama, Dana Marsetiya. (2019). An Effective Hybrid Sine Cosine Algorithm to Minimize Carbon Emission on Flow-shop Scheduling Sequence Dependent Setup. Jurnal Teknik Industri, 20(1), 62-72. doi:https://doi.org/10.22219/JTIUMM.Vol20.No1.62-72