International Journal of Technology 10(2): 320-331
ISSN 2086-9614 © IJTech 2019

A NEW HYBRID METAHEURISTICS ALGORITHM FOR MINIMIZING ENERGY CONSUMPTION IN THE FLOW SHOP SCHEDULING PROBLEM

Dana Marsetiya Utama[1*], Dian Setiya Widodo[2], Wahyu Wicaksono[1], Leo Rizky Ardiansyah[2]

[1]*Department of Industrial Engineering, University of Muhammadiyah Malang, Jl. Raya Tlogomas 246 Malang, 65144, Indonesia*
[2]*Department of Manufacturing Technology, Vocational Faculty, University of 17 Agustus 1945 Surabaya, Jl. Semolowaru No. 45 Surabaya, 60118, Indonesia*

(Received: July 2018 / Revised: September 2018 / Accepted: February 2019)

ABSTRACT

In this study, we discuss the problem of permutation flowshop scheduling problem (PFSP) to reduce total energy consumption (TEC). We offer a new hybrid meta-heuristic algorithm for solving the problem. The paper aims to combine the cross entropy and genetic algorithm (CEGA) with the simulated annealing (SA) algorithm. The CEGA is applied to find the best initial solution inside the SA algorithm and the proposed algorithm is compared to previous tests of the famous NSGA-II and GA-SA algorithm. During study of the numerical test, the proposed algorithm genuinely useful is compared certain efficient algorithms of the from previous research.

Keywords: Algorithm; Energy consumption; Flow shop, Meta-heuristic

1. INTRODUCTION

Recently, Total Energy Consumption (TEC) in the manufacturing sector has received much attention from experts. This has been focused on highly TEC in the manufacturing sector. TEC in this sector requires almost half of the total energy needs in country. In the USA, it requires 33% of the total electricity of the country (Evans, 2003), while in Germany it requires 47% of electricity from all energy requirements (Dai et al., 2013). The electricity consumption of the sector needs fossil fuels for electricity generation; therefore, experts consider such consumption to be a problem because of the decreasing availability of these fuels. Some experts have made efforts to minimize TEC, one of which is scheduling, which refers to the arrangement of resources (machines) to complete the job (Surjandari et al., 2015). Generally, the goal of scheduling is to minimize completion time (Thawongklang & Tanwanichkul, 2016). However, some experts are now using scheduling to reduce TEC.

Several researchers have researched flow shop scheduling problems to reduce TEC. Zhang et al. (2014), Brundage et al. (2014) and Zanoni et al. (2014) have succeeded in minimizing TEC in simple flow shop problems, using a heuristic algorithm as a solution. Besides, heuristic algorithms are explicitly used to solve specific problems. In recent years, some meta-heuristic algorithm have also been used to solve the classic flow shop problem in order to minimize TEC. These algorithms include simulated annealing (SA) (Iqbal & Al-Ghamdi, 2018); a genetic algorithm (GA) (Liu et al., 2017); and particle swarm optimization (PSO) (Tang et al., 2016). In hybrid flow shop problems, several studies to minimize TEC have been conducted by Luo et al. (2013), Dai et al. (2013) and Liu and Huang (2014), who used meta-heuristic algorithms to solve

[*]Corresponding author's email: dana@umm.ac.id, Tel. +62-341-464318, Fax. +62-341-460435
Permalink/DOI: https://doi.org/10.14716/ijtech.v10i2.2194

energy consumption problems. In this article, we focus on the Permutation Flow-Shop Scheduling Problem (PFSP). Researchers claim that a solution to this problem cannot be found in polynomial time. Therefore, PFSP is considered an NP-Hard problem (Garey et al., 1976; Sayadi et al., 2010). Because of the importance of this problem, several efforts have been made by experts to develop algorithms to minimize TEC.

In recent years, SA, Cross-entropy (CE) and GA algorithms have been used to solve scheduling problems. The SA algorithm is a meta-heuristic algorithm, which were first introduced by Kirkpatrick et al. (1983) for optimization. However, this algorithm is now used in most PFSP scheduling problems (Pinedo, 2016). Like the SA, GA is also a meta-heuristic algorithm based on mimicking natural selection and recombination (Holland, 1992). CE is another meta-heuristic algorithm applied to rare event simulations, continuous optimization, and combinatorial optimization (Deng, 2006). This algorithm is useful in solving complex combinatorial optimization problems (De Boer et al., 2005). In recent years, some experts have used meta-heuristic algorithms to solve PFSP, and some simple meta-heuristics have been applied to reduce TEC. However, classic meta-heuristics need a long time if used in large cases (Santosa et al., 2011). Recently, some hybrid meta-heuristic alternatives have been developed to solve PFSP. These algorithms include a hybrid GA with SA (Dai et al., 2013); a hybrid GA with TS (Sukkerd and Wuttipornpun, 2016); a hybrid of ABC and TS (Li and Pan, 2015); and a hybrid of CE and GA (Santosa et al., 2011).

Although many hybrid meta-heuristic algorithms have been developed to solve PFSP problems, they still display certain weaknesses, namely the long computing time for large-scale problems and optimal local solutions. Although they do need a long computation time, hybrid meta-heuristics give better performance compared to simple meta-heuristics. Many meta-heuristic algorithms have good global search capabilities, while some have local search capabilities. At present, few papers focus on minimizing TEC in PFSP. To our knowledge, none integrate CE and GA (CEGA) with SA. Therefore, this paper aims to combine CEGA with SA to reduce TEC, an approach we term CEGASA. This algorithm follows the rules for fixed energy consumption (FEC) Li et al. (2011). Hence, the paper focuses on minimizing TEC by following FEC rules. The remainder of this paper is organized as follows: Part 2 explains problem discription, example problem, proposes algorithms, and describes the experimental procedure. Section 3 then presents the computational experiments, experimental parameters, and comparison algorithms. Finally, the the conclusion is made in section 4.

2. METHODS

2.1. Problem Description

In the PFSP problem, there are n jobs completed on m machines, that which are arranged in the same order. The problem aims to schedule every job on each machine in order to minimize energy consumption. Some assumptions of PFSP are that: (1) the group of jobs is prepared on machines in the same sequence; (2) every machine can process one task in each period; (3) job preemption is banned; (4) every job is ready in period $t = 0$; (5) a job will begin on machine j only if it has completed the process on machine j-1; (6) there is no precedence relationship between jobs; (7) every machine starts in period $t = 0$; (8) every machine stops when the last job on it has been completed (every machine stops independently of other machines); and (9) setup time is covered in the processing time. The notation in the total energy consumption used in this article is as follows:

i : index of jobs, $i = 1, 2, \ldots, n$
j : index of machines, $j = 1, 2, \ldots, n$
n : total number of jobs

m	:	total number of machines
$P_{i,j}$	:	processing time of job sequence i on machines j
P_{ej}	:	energy consumption index of machine j
I_{ej}	:	energy consumption index of machine j when idle
C_{ij}	:	completion time of job sequence i at on machines j
T_j	:	completion time of machines j
B_j	:	total busy time of machines j
I_j	:	total idle time of machines j
TEC	:	total energy consumption

Based on the above notations, the objective function of this PFSP problem is to minimize total energy consumption (*TEC*) (Li et al., 2018a; Li et al., 2018b). Furthermore, the following is the formula of the PFSP problem:

$$C_{1,1} = P_{1,1} \tag{1}$$

$$C_{1,j} = C_{1,j-1} + P_{1,j}\ , \quad j = 2\,..\,m \tag{2}$$

$$C_{i,1} = C_{i-1,1} + P_{i-1,1}\ , \; i = 2\,..\,n \tag{3}$$

$$C_{i,j} = \max\left(C_{i-1,j}, C_{i,j-1}\right) + P_{i,j}\ , \qquad i = 2\,..\,n, \qquad j = 2\,..\,m \tag{4}$$

$$Bj = \textstyle\sum_{i=1}^{n} P_{i,j}\ , \qquad \forall\, j = 1..m \tag{5}$$

$$Tj = \max(C_{i,j})\ , \qquad \forall\, i = 1\,..\,n,\ j = 1..m \tag{6}$$

$$Ij = Tj - Bj\ , \qquad \forall\, j = 1..m \tag{7}$$

$$TEC = \textstyle\sum_{j=1}^{m}(Bj.Pej + Ij.Iej) \tag{8}$$

The PFSP model is modified from Li et al. (2018b). The best permutations are defined as those which have the minimum TEC. The PFSP model to minimize energy consumption is as follows:

$$\text{Objective function } Z = \min TEC \tag{9}$$

Subject to :

$$\left.\begin{aligned}
&C_{1,1} = P_{1,1} \\
&C_{1,j} = C_{1,j-1} + P_{1,j}\ , \qquad j = 2\,..\,m \\
&C_{i,1} = C_{i-1,1} + P_{i-1,1}\ , \qquad i = 2\,..\,n \\
&C_{i,j} = \max\left(C_{i-1,j}, C_{i,j-1}\right) + P_{i,j}\ , \quad i = 2\,..\,n, \qquad j = 2\,..\,m \\
&Bj = \textstyle\sum_{i=1}^{n} P_{i,j}\ , \qquad \forall\, j = 1..m \\
&Tj = \max(C_{i,j})\ , \quad \forall\, i = 1\,..\,n,\ j = 1..m \\
&Ij = Tj - Bj\ , \quad \forall\, j = 1..m \\
&TEC = \textstyle\sum_{j=1}^{m}(Bj.Pej + Ij.Iej)
\end{aligned}\right\} \tag{10}$$

Equation 1 describes the completion time of sequence job one on machine *1*; Equation 2 describes that of machines 2 to *m*; Equation 3 describes the completion time of sequence job *i* of machine *1*; Equation 4 shows that of machine *j*; Equation 5 describes the total busy time of machines *j*; Equation 6 shows the completion time of machines *j* of the permutation; Equation 7 shows the total idle time of machines *j* of the permutation; Equation 8 describes the *TEC* of the permutation (the objective function); Equation 9 describes the objective function of the PFSP model to minimize energy consumption; and Equation 10 describes constraint of the PFSP model to minimize energy consumption. This constraints in the model are Equations 1 to 8.

2.2. Example Problem

As an example problem, there are three jobs and three machines (see Table 1). Figure 1 shows the completion time of each job if sequences J1, J3, J2 (a) are 6, 9 and 11. Based on Equation 1, the calculation of energy consumption is 42 W. However, in sequences J2, J3, J1 (b), the completion times of each job are 6, 9 and 11, with an energy consumption of 43 W. Two schedules (a and b) can be selected as the best solution if the goal of the scheduling is to minimize completion time. Although they have the same completion time, if the aim is to minimize TEC, sequence (a) shows the better result. Total idle time for sequence (a) is 10, while for sequence (b) it is 11. The total idle time and FEC for sequence (a) are lower than for (b). The idle time of each machine influences the variation in the TEC in flow shop problems. Hence, total energy consumption in sequence (a) is lower than in sequence (b).

Table 1 Processing time, *Pej*, and *Iej* of each machine (minutes)

Job	Machine		
	M1	M2	M3
J1	3	2	1
J2	3	1	2
J3	2	1	3
Pej	2	1	2
Iej	1	1	1

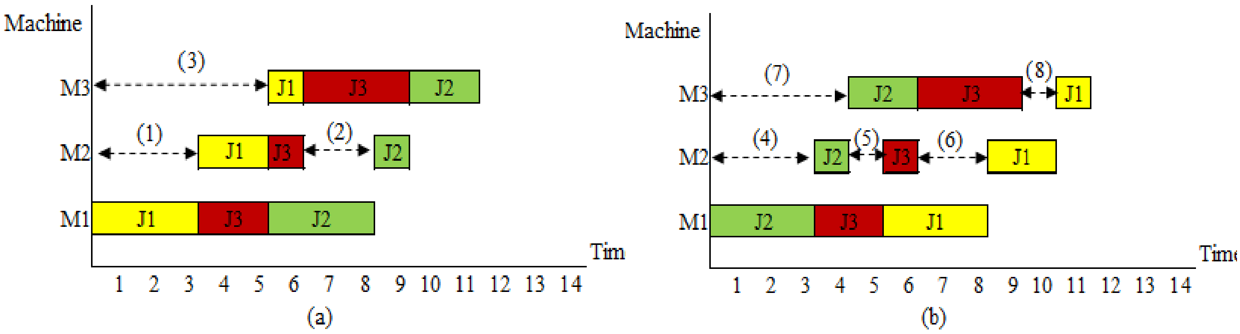

Figure 1 Difference between the two sequences in completion time

2.3. Proposed Algorithm

We propose the CEGASA algorithm to minimize energy consumption. Figure 2 shows an illustration of the algorithm proposed to minimize TEC in PFSP problems. It uses an integrated CEGA algorithm as the initial solution in SA to minimize energy consumption, which is modified from Santosa et al. (2011). Details of the CEGA stages can be seen in Algorithm 1. Notations of the algorithm are as follows:

LFR	:	Linear Fitness Ranking
I	:	state job index in the sample matrix
N	:	number of samples
x	:	declared sample *x* (between 1 and *N*)
TECmax	:	states the maximum fitness value of the sample, which is equal to 1 / *TEC*
TECmin	:	states the minimum fitness value of the sample, which is equal to 1 / *TEC*
Pps	:	crossover parameter
α	:	coefficient of fineness ($0<\alpha<1$)
u	:	updated value of the crossover parameter

$\overline{TECe}$: average energy consumption of the sample
TECbest : minimum energy consumption of the sample
Pm : mutation parameter

Some of the equations in the CEGA algorithm are as follows:

$$LFR(I(N-x+1)) = TECmax - (TECmax - TECmin) * ((x-1)/(N-1)) \tag{11}$$

$$Pps(x) = (1-\alpha) * u + (Pps(x+1) * \alpha) \tag{12}$$

$$u = \frac{\overline{TECe}}{2*TECbest} \tag{13}$$

$$Pm = \frac{Pps}{2} \tag{14}$$

Equation 11 describes the formula for the weighting of elite samples use Linear Fitness Ranking (*LFR*). The *LFR* for the present iteration is computed from the fitness value of all the samples generated in the previous iteration. The fitness value is same as 1/*TEC*. *x* is stated in the *x*-th sample (which is valued between 1 and *N*), and *I* states the job index of the sample matrix (Santosa et al., 2011). Equation 12 shows the updated formula crossover parameter for sample *x*, while Equation 13 describes the formula for the updated value of crossover parameter. Equation 14 describes the formula the of the updated mutation parameter. The termination criterion of the CEGA is if $Pps \leq \beta$.

Algorithm 1 Proposed CEGA Procedure:

1. *Determine the number of samples (N), the parameter of the ρ, the coefficient of fineness (α), the crossover parameter (Pps), and the termination criterion (β)*
2. *Generate random permutation N from the samples*
3. *Calculate the objective function based on equation (8) for N samples*
4. ***While** Pps > β*
5. *Determine the number of sample elites (ρ*N)*
6. *Determine the weighting weighting of the elite samples and calculate Linear Fitness Ranking (LFR) based on equation 11*
7. *Update crossover parameter (Pps) and mutation parameter (Pm) as in equations 12, 13, and 14*
8. *Select parents used as a roulette wheel selection*
9. *Crossover*
10. *Mutation*
11. *Return to step 3*
12. ***end of while***

The simulated annealing (*SA*) algorithm imitates the way the steel is heated at specific temperatures (*Tmax*) and then cooled slowly to appropriate temperatures (*Tmin*). This procedure is used to obtain the appropriate form of steel. Based on which similarity, SA can be implemented in optimization problems (Kirkpatrick et al., 1983). The notations of the proposed SA algorithm are as follows;

T : temperature
t : iteration
λ : reduction factor
Tmax : maximum temperature at initialization
Tmin : maximum temperature (termination criteria)
Trymax : number of iterations at each temperature *T*

In this paper, the initial solution of the *SA* algorithm is based on the *CEGA* algorithm (Algorithm 1). Based on this, the *SA* algorithm produces a new solution at each temperature. At temperature *T*, *SA* generates a new permutation based on neighboring solution generation (generating a new solution by pairwise interchange) and calculates a new *TEC*'. Generation of a new solution by pairwise interchange is based on Mirsanei et al. (2011). Furthermore, the previous solution and new TEC are compared. The solution deviation of the iteration is expressed as $\Delta E = TEC' - TEC$. The new solution to the algorithm is received by SA if the value of $\Delta E \leq 0$. New solutions are also accepted if the value $exp^{-\frac{\Delta E}{T}} \geq r$ (r is a random number with a range $r \in [0,1]$). Otherwise, the new solution is rejected by the algorithm. At T temperature, the SA carries out solution search as much as $trymax$ iteration. Furthermore, T temperature decreases at each iteration, and the temperature decrease factor is expressed as λ. In this paper, the decrease in temperature in the iteration $(t + 1)$ is made using $T(t+1) = \lambda.T(t)$. The coefficient value λ is a value between 0 and 1. This procedure was repeated continuously until the termination criteria were met ($T \leq Tmin$). The Simulated Annealing algorithm is shown in Algorithm 2.

Algorithm 2 Proposed Simulated Annealing algorithm

1. ***Initialization*** *: select* $Tmax, Tmin, \lambda, Trymax$ *and initialize temperature at* $T = Tmax$
2. *An initial solution to the permutation and TEC is selected based on algorithm 1 (proposed CEGA algorithm)*
3. ***While*** $T \geq Tmin$
4. ***While*** *the number of* $attempts \leq Trymax$
5. *Create permutation based on neighboring solution generation (generate a new solution by pairwise interchange) and calculate new TEC' based on equation 8*
6. ***if*** $TEC' \leq TEC$
7. *receive the new solution and replace the previous solution with the new one*
8. ***else if*** $TEC' > TEC$
9. *let* $\Delta E = TEC' - TEC$, *create a random range* $r \in [0,1]$, *accept if* $exp^{-\frac{\Delta E}{T}} \geq r$ *replace the previous solution with the new one*
10. ***end of if***
11. ***end of while***
12. *Decrease the temperature* $T(t+1) = \lambda.T(t)$
13. ***end of while***

2.4. Experimental Procedure

To evaluate the proposed CEGASA algorithm, several examples of problems were made randomly. The data needed to establish the algorithm consist of the number of jobs, the correct time range, the number of stages, the energy distribution range of each machine, and the energy range at idle of each machine. We conducted experiments for three job families: small (5, 10 and 20), medium (25, 50 and 75), and large (100, 150 and 200). Therefore, the total variation in the number of jobs in the experiment is 9. Number of jobs included 5, 10, and 20. Processing time was generated by the uniform distribution (20, 200) in minutes, while the load fix power machine was generated by the uniform distribution (1, 100) of each machine in watts (W). In addition, energy consumption of machines when idle was produced by the uniform distribution (1, 10) of each machine in watts (W). We conducted experiments for all combinations of the different number of jobs and machines.

In metaheuristics, algorithm parameters influence the objective function. In this section, the behaviors of various parameter levels in the proposed CEGASA are studied. Experiments were conducted with a combination of different parameters; those used were α, initial *T* and *λ*. All the experimental parameters were at two levels. Each considered parameter produced a total of 2×2×2 = 8 various CEGASA algorithms. We used parameter $N = 10$, $\rho = 0.2$; $Trymax = 10$, $\beta =$

0.0001; and *Pps* = 1. Furthermore, α used 0.2 and 0.9, initial T used 100 and 500, and λ used 0.2 and 0.8. Simulation experiments were carried out for each parameter, with 216 experiments conducted for the three parameters (two levels), nine job variations, and three variations in the number of machines. The experiments were conducted to ascertain the best parameters of the CEGASA algorithm.

Furthermore, the parameter level chosen for the experiment was based on total energy consumption and computational time. The best parameters from the experimental results were used for comparison with several other algorithms. In addition, evaluation was made of the effectiveness of the proposed CEGASA algorithm using the Energy Consumption Ratio (ECR) and the Wilcoxon Test.

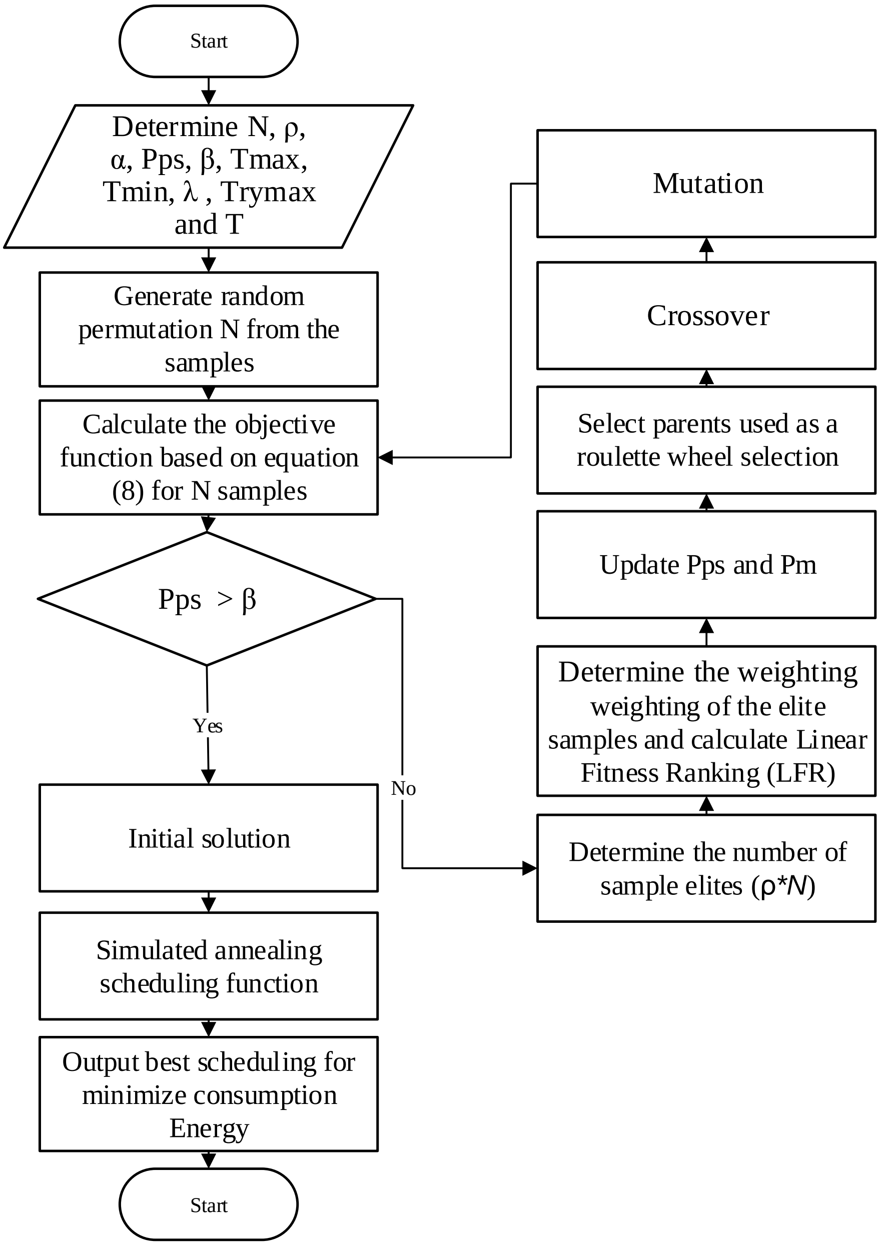

Figure 2 Illustration of the proposed CEGASA algorithm

3. RESULTS AND DISCUSSION

3.1. Computational Experiments

In this section, a CEGASA experiment was conducted to establish the feasibility and effectiveness of the algorithm. Numerical tests were made with Matlab R16 software run on Windows 8.1 processors AMD x86-64 RAM 4 GB. Some numerical tests used several manufacturing environment cases.

3.2. Parameters Experiment

The results of the numerical test and algorithm parameters of the CEGASA proposed are shown in Table 1. The best solution is obtained if the coefficient of fineness (α) value is the most significant and the other parameters are constant. The findings of the coefficient of fineness (α) correspond to Deng (2006). If the temperature (T) value increases and the other parameters remain constant, a better solution is obtained. Moreover, if the reduction factor (λ) value decreases and the other parameters are constant, a better solution is also obtained. The findings of temperature (T) and the reduction factor (λ) correspond to Haddock and Mittenthal (1992). The computational time simulation parameters can be seen in Table 2. If T and the coefficient of fineness (α) value are significant when the reduction factor (λ) is small, long computational time is needed. Otherwise, if T and the coefficient of fineness (α) are low when the reduction factor (λ) is significant, less time is needed. The better the solution obtained, the higher the computational time.

Table 1 Comparison of each parameter with regard to energy consumption (in watts)

α	Job Family	n job	m machine	T=100		T=1000	
				λ =0.2	λ =0.8	λ =0.2	λ =0.8
0,2	Small	5	5	164271	164535	164379	164535
		10	10	564314	560504	563324	564334
		20	20	2209627	2218468	2210803	2198385
	Medium	25	5	915584	916332	915824	916988
		50	10	2801429	2808184	2808129	2809554
		75	20	8194809	8211889	8203286	8205610
	High	100	5	3623926	3624444	3622128	3622430
		150	10	8574852	8579187	8567037	8569517
		200	20	20749953	20750686	20713867	20715287
0,8	Small	5	5	164349	164271	164349	164271
		10	10	562304	560749	560779	559969
		20	20	2182453	2196096	2198483	2180577
	Medium	25	5	915512	916324	914834	916170
		50	10	2799539	2800569	2803509	2803274
		75	20	8186199	8195292	8187004	8192737
	High	100	5	3622320	3622432	3622140	3622504
		150	10	8568337	8570837	8566852	8569197
		200	20	20740481	20740452	20731582	20732269

3.3. Comparison Algorithms and Evaluation

The performance of the algorithms is measured by the Energy Consumption Ratio (ECR). The ECR is defined as the Energy Consumption (EC) of the proposed CEGASA algorithm divided by EC different algorithms (Equation 15). The CEGASA has a higher performance than other algorithms if ECR < 1, but it has the same performance if it produces an ECR value = 1. Moreover, other algorithm has a higher performance if ECR > 1. The CEGASA is compared with algorithms including GA-SA (Dai et al., 2013) and NSGA-II (Li et al., 2018b).

$$ECR = \frac{\text{EC proposed algorithm}}{\text{EC other algorithm}} \tag{15}$$

Table 2 Comparison of the computation time of each parameter (in seconds)

α	Job Family	n job	m machine	T=100 λ =0.2	T=100 λ =0.8	T=1000 λ =0.2	T=1000 λ =0.8
0.2	Small	5	5	1.6278	0.6875	1.3021	5.5340
		10	10	2.3875	5.5340	1.7361	6.0764
		20	20	3.5806	1.0938	3.2556	1.2344
	Medium	25	5	4.2319	1.2500	3.5806	1.3281
		50	10	6.5104	2.0156	5.4257	2.3906
		75	20	1.4531	2.9375	1.3594	3.1563
	High	100	5	1.9688	3.1094	1.8906	3.6094
		150	10	3.1406	5.2031	3.1875	5.5938
		200	20	4.8125	8.1563	5.2656	8.5938
0.8	Small	5	5	3.9063	1.0625	4.0146	1.1094
		10	10	5.9681	1.3125	6.1847	1.3125
		20	20	1.4531	2.1094	1.3750	2.0000
	Medium	25	5	1.6731	2.2969	1.7344	2.4688
		50	10	2.4688	3.7031	2.6094	3.5781
		75	20	3.6094	5.1563	3.5156	5.4063
	High	100	5	4.5000	5.9531	4.6250	6.1563
		150	10	6.5938	9.4375	7.1406	9.4063
		200	20	9.6563	12.3750	9.9531	13.3281

A comparison of the algorithm energy consumption and the ratio of the CEGASA between some other algorithms was made (see Table 3). The ECR results show the average ECR NSGA-II (Li et al., 2018b) values are significant at 0.88 and 0.93 for GA-SA (Dai et al., 2013). Table 4 shows that NSGA-II (Li et al., 2018b) needed the shortest computational time in every case studied; however, it did not represent the best solution (see Table 3). Several numerical tests showed that the CEGASA performances were better. Furthermore, a comparison of the algorithm was also made with the Wilcoxon test (Table 5). These test results on CEGASA ECR performance showed significant differences in the performance of the ECR GASA and NSGA-II.

Table 3 Comparison of the algorithms for energy consumption and ECR

Job Family	n job	m machine	Energy Consumption (in watts)			ECR		
			CEGASA	NSGA-II	GA-SA	CEGASA	NSGA-II	GA-SA
Small	5	5	164349	182610	180603	1	0.90	0.91
	10	10	562304	661534.1	611200	1	0.85	0.92
	20	20	2182453	2452194	2372232	1	0.89	0.92
Medium	25	5	915512	1077073	943827	1	0.85	0.97
	50	10	2799539	3293575	2978233	1	0.85	0.94
	75	20	8186199	9518836	8802365	1	0.86	0.93
High	100	5	3622320	4116273	3853532	1	0.88	0.94
	150	10	8568337	9520374	9019302	1	0.90	0.95
	200	20	20740481	23044979	22301592	1	0.90	0.93
Average Energy Consumption Ratio						1	0.88	0.93

Table 4 Comparison of the computation time for some algorithms

Job Family	n job	m machine	Computation time (in seconds)		
			CEGASA	NSGA-II	GA-SA
Small	5	5	3.9063	3.5156	3.5547
	10	10	5.9681	5.0728	5.4906
	20	20	1.4531	1.2933	1.3369
Medium	25	5	1.6731	1.4221	1.6229
	50	10	2.4688	2.0985	2.3207
	75	20	3.6094	3.1041	3.3567
High	100	5	4.5000	3.9600	4.2300
	150	10	6.5938	5.9344	6.2641
	200	20	9.6563	8.6907	8.9804

Table 5 Wilcoxon test of the ECR

Test	Z	Asymp. Sig. (2-tailed)
CEGASA - NSGA-II (Li et al., 2018b)	-2.684	0.007
CEGASA - GA-SA (Dai et al., 2013)	-2.673	0.008

4. CONCLUSION

We have discussed the problem of PFSP in reducing energy consumption and offer the CEGASA algorithm to solve this problem. The algorithm has been compared with other algorithms and numerical experiments have proven that it achieves optimum energy consumption. Some other research areas could be studied in future work. We propose that the CEGASA be used as an initial solution for other meta-heuristic algorithms, and ultimately be applied to the reduction of energy consumption in more complex PFSPs.

5. ACKNOWLEDGEMENT

The authors would like to thank the Directorate of the Research University of Muhammadiyah Malang for support in conducting the research. We would also like to thank the Department of Industrial Engineering Optimation Laboratory for use of their facilities.

6. REFERENCES

Brundage, M.P., Chang, Q., Li, Y., Xiao, G., Arinez, J., 2014. Energy Efficiency Management of an Integrated Serial Production Line and HVAC System. *IEEE Transactions on Automation Science and Engineering*, Volume 11(3), pp. 789–797

Dai, M., Tang, D., Giret, A., Salido, M.A., Li, W.D., 2013. Energy-efficient Scheduling for a Flexible Flow Shop using an Improved Genetic-simulated Annealing Algorithm. *Robotics and Computer-Integrated Manufacturing*, Volume 29(5), pp. 418–429

De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y., 2005. A Tutorial on the Cross-entropy Method. *Annals of Operations Research*, Volume 134(1), pp. 19–67

Deng, L.Y., 2006. The Cross-entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning. *Technometrics*, Volume 48(1), pp. 147–148

Evans, L.B., 2003. Saving Energy in Manufacturing with Smart Technology. *World*, Volume 6(2), pp. 112–120

Garey, M.R., Johnson, D.S., Sethi, R., 1976. The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, Volume 1(2), pp. 117–129

Haddock, J., Mittenthal, J., 1992. Simulation Optimization using Simulated Annealing. *Computers & Industrial Engineering,* Volume 22(4), pp. 387–395

Holland, J.H., 1992. Genetic Algorithms. *Scientific American,* Volume 267(1), pp. 66–73

Iqbal, A., Al-Ghamdi, K.A., 2018. Energy-efficient Cellular Manufacturing System: Eco-friendly Revamping of Machine Shop Configuration. *Energy,* Volume 163, pp. 863–872

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science,* Volume 220(4598), pp. 671–680

Li, J., Pan, Q., 2015. Solving the Large-scale Hybrid Flow Shop Scheduling Problem with Limited Buffers by a Hybrid Artificial Bee Colony Algorithm. *Information Sciences,* Volume 316, pp. 487–502

Li, J., Sang, H., Han, Y., Wang, C., Gao, K., 2018a. Efficient Multi-objective Optimization Algorithm for Hybrid Flow Shop Scheduling Problems with Setup Energy Consumptions. *Journal of Cleaner Production,* Volume 181, pp. 584–598

Li, S., Liu, F., Zhou, X., 2018b. Multi-objective Energy-saving Scheduling for a Permutation Flow Line. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Volume 232(5), pp. 879–888

Li, W., Zein, A., Kara, S., Herrmann, C., 2011. An Investigation into Fixed Energy Consumption of Machine Tools. *In:* Hesselbach J., Herrmann C. (eds) Glocalized Solutions for Sustainability in Manufacturing. Springer, Berlin, Heidelberg, pp. 268–273

Liu, C., Huang, D., 2014. Reduction of Power Consumption and Carbon Footprints by Applying Multi-objective Optimisation via Genetic Algorithms. *International Journal of Production Research,* Volume 52(2), pp. 337–352

Liu, G., Zhou, Y., Yang, H., 2017. Minimizing Energy Consumption and Tardiness Penalty for Fuzzy Flow Shop Scheduling with State-dependent Setup Time. *Journal of Cleaner Production,* Volume 147, pp. 470–484

Luo, H., Du, B., Huang, G., Chen, H., Li, X., 2013. Hybrid Flow Shop Scheduling Considering Machine Electricity Consumption Cost. *International Journal of Production Economics,* Volume 146(2), pp. 423–439

Mirsanei, H., Zandieh, M., Moayed, M.J., Khabbazi, M. R., 2011. A Simulated Annealing Algorithm Approach to Hybrid Flow Shop Scheduling with Sequence-dependent Setup Times. *Journal of Intelligent Manufacturing,* Volume 22(6), pp. 965–978

Pinedo, M.L., 2016. *Scheduling Theory, Algorithm, and Systems*. Springer International Publishing, New York

Santosa, B., Budiman, M.A., Wiratno, S.E., 2011. A Cross Entropy-Genetic Algorithm for m-Machines No-wait Job-shop Scheduling Problem. *Journal of Intelligent Learning Systems and Applications,* Volume 3(03), pp 171–180

Sayadi, M., Ramezanian, R., GhaffariNasab, N., 2010. A Discrete Firefly Meta-heuristic with Local Search for Makespan Minimization in Permutation Flow Shop Scheduling Problems. *International Journal of Industrial Engineering Computations,* Volume 1(1), pp. 1–10

Sukkerd, W., Wuttipornpun, T., 2016. Hybrid Genetic Algorithm and Tabu Search for Finite Capacity Material Requirement Planning System in Flexible Flow Shop with Assembly Operations. *Computers & Industrial Engineering,* Volume 97, pp. 157–169

Surjandari, I., Rachman, A., Dhini, A., 2015. The Batch Sheduling Model for Dynamic Multiitem, Multilevel Production in an Assembly Job-shop with Parallel Machines. *International Journal of Technology,* Volume 1, pp. 84–96

Tang, D., Dai, M., Salido, M.A., Giret, A., 2016. Energy-efficient Dynamic Scheduling for a Flexible Flow Shop using an Improved Particle Swarm Optimization. *Computers in Industry,* Volume 81, pp. 82–95

Thawongklang, K., Tanwanichkul, L., 2016. Application of Production Scheduling Techniques for Dispatching Ready-mixed Concrete. *International Journal of Technology,* Volume 7(7), pp. 1163–1170

Zanoni, S., Bettoni, L., Glock, C.H., 2014. Energy Implications in a Two-stage Production System with Controllable Production Rates. *International Journal of Production Economics,* Volume 149, pp. 164–171

Zhang, H., Zhao, F., Fang, K., Sutherland, J.W., 2014. Energy-conscious Flow Shop Scheduling under Time-of-Use Electricity Tariffs. *CIRP Annals-Manufacturing Technology,* Volume 63(1), pp. 37–40