JURNAL ILMIAH TEKNIK INDUSTRI
ISSN: 1412-6869 (Print), ISSN: 2460-4038 (Online)
Journal homepage: http://journals.ums.ac.id/index.php/jiti/index
doi:10.23917/jiti.v19i1.10079

Energy-Efficient Flow Shop Scheduling Using Hybrid Grasshopper Algorithm Optimization

Dana Marsetiya Utama[1a◆], Teguh Baroto[1b], Dian Setiya Widodo[2c]

Abstract. *Manufacturing companies have a significant impact on environmental damage, and manufacturing companies' energy consumption is a widespread issue because the energy used is derived from fossil fuels. This research aims to minimize energy consumption using develop Hybrid Grasshopper Algorithm Optimization (HGAO). The focus of the issue in this article is the Permutation Flow Shop Scheduling Problem (PFSSP). A case study was conducted in offset printing firms. The results showed that the HGAO algorithm is capable of reducing energy consumption in offset printing firms. The higher the population of search agents and iterations produces less energy consumption. The HGAO algorithm is also compared with the genetic algorithm (GA). The results show that HGAO is more efficient in reducing energy consumption than GA.*

Keywords: *scheduling, flow shop, energy consumption, hybrid grasshopper algorithm optimization.*

I. INTRODUCTION

Recently, manufacturing companies have had a significant impact on environmental damage (Dai et al., 2013; Maulana et al., 2019; Widodo & Utama, 2019), and electrical energy is the primary source of energy in the manufacturing sector (Fang et al., 2011). Energy consumption is becoming an essential issue in manufacturing companies by making it environmentally sustainable (Jiang et al., 2018). Electrical energy is primarily derived from fossil fuels. The higher energy requires that a company needs, the greater its fossil fuel needs, and it harms the environment. Scheduling plays an essential role in reducing energy consumption (Grobler et al., 2010). One of the scheduling problems is the Permutation Flow Shop Scheduling Problem (PFSSP) (Utama, 2018a; Utama et al., 2020). Many experts claim that the PFSSP case can not be resolved in polynomial time (Utama, Ardiansyah, & Garside, 2019; Utama, Garside, & Wicaksono, 2019). Therefore, PFSSP is included in the NP-Hard problem (Garey et al., 1976). One of the techniques for reducing consumption in the manufacturing sector is the use of energy-efficient machinery (Elias et al., 2019; Utama, 2019a). However, it requires very high costs and is not owned by the small and medium-sized manufacturing industries (Tian et al., 2018). Therefore, scheduling is one strategy for reducing energy consumption (Utama, Widodo, Wicaksono, & Ardiansyah, 2019). The scheduling problem has attracted much attention to researchers. Researchers have previously referred to this issue as Energy Efficient Scheduling (Gong et al., 2020; Öztop et al., 2020).

In recent years, metaheuristic algorithms have been used to find optimal solutions to scheduling problems (Pan et al., 2017; Widodo et al., 2014). Some previous studies use heuristics (Utama, 2018b) and metaheuristic algorithms to minimize energy consumption in scheduling cases. Popular metaheuristics algorithms for reducing energy consumption include particle swarm optimization (PSO) (Tang et al., 2016), Salp Swarm Algorithm (SSA) (Utama, 2019b), memetic differential evolution algorithm (DE) (Wu & Che, 2019), Genetic Algorithm (GA) (Liu et al., 2016; Piroozfard et al., 2018), Whale Optimization Algorithm (WOA) (Jiang et al., 2018), and

[1] Department of Industrial Engineering, Faculty of Engineering, Universitas Muhammadiyah Malang, Jl.Raya Tlogomas No. 246, Tlogomas, Kota Malang, Jawa Timur 65144
[2] Department of Manufacturing Technology, Faculty of Vocational Education, Universitas 17 Agustus 1945 Surabaya, Jl. Semolowaru No. 45 Surabaya, Jawa timur, Indonesia, 60118

[a] email: dana@umm.ac.id
[b] email: teguh@umm.ac.id
[c] email: diansetiyawidodo@untag-sby.ac.id
◆ corresponding author

Submited: 31-01-2020 Revised: 03-06-2020
Accepted: 07-06-2020

teaching-learning-based optimization algorithm (Lei et al., 2018). Furthermore, the researchers create more advanced algorithms, including a hybrid multi-objective backtracking algorithm (Lu et al., 2017), Hybrid Genetic Algorithm (HGA) (Liu et al., 2019), improved genetic-simulated annealing algorithm (SA-GA) (Dai et al., 2013), and collaborative optimization algorithm (COA) (Chen et al., 2019). Certain methods that are capable of reducing total energy consumption are the heuristic algorithm (Brundage et al., 2013; Shrouf et al., 2014; Zanoni et al., 2014; Zhang et al., 2014) and the combination of a metaheuristic algorithm (Utama et al., 2019; Utama, Baroto, Maharani, Jannah, & Octaria, 2019).

Based on previous research, there is no research using the Hybrid Grasshopper Algorithm Optimization (HGAO) approach to minimize energy consumption. Therefore, one of the ways to overcome this problem of energy consumption is to use HGAO. HGAO is an algorithm inspired by the behavior of grasshoppers in nature (Saremi et al., 2017). The researchers have developed a grasshopper algorithm by incorporating local search procedures. This research aims to minimize energy consumption in the PFSSP problem using a Hybrid Grasshopper Algorithm Optimization (HGAO).

II. RESEARCH METHOD

In this section, we explain the PFSSP problem's assumptions, the definition of the problem, the Hybrid Grasshopper Algorithm Optimization procedure, and the method of collect and experiment data.

Assumptions of the Problem

In this PFSSP problem, some assumes used such as: (1) n jobs (n= 1,2,3... I operated in the same order on a sequence of m machines (m= 1,2,3.. j). (2) Pij processing time is the completion time of the job sequence i function on machines j. (3) All computers are available at t= 0, (4) the setup time includes the operation time. (5) The time of removal includes processing time. (6) Every job that starts processed until it is done (no pre-emption). (8) every machine stops when the last job on each machine is finished (each machine that stops independently of other machines). (9) every machine starts actively when the first operation is available. The evaluation of the question of energy consumption in this article is defined as follows:

i	:	Job Index, i = 1, 2, …, n
j	:	Machinery Index, j = 1, 2, …, n
n	:	Total jobs
m	:	Total machines
$P_{i,j}$	:	Processing time of the job sequence i on the machines j
P_{ej}	:	Index of energy consumption of the machine j
I_{ej}	:	Index of energy consumption of machine j when idle
C_{ij}	:	Completion time of the job sequence i on the machines j
T_j	:	Completion time from machines j
B_j	:	Total busy time from machines j
I_j	:	Total idle time from machines j
TEC	:	Total Energy Consumption

Definition of Problem

The problem of energy efficiency in the PFSSP model is modified from S. Li, Liu, and Zhou (2018). The best scheduling is defined as having a minimum TEC. The PFSSP model for minimizing energy consumption is as follows:

Objective function Z = min TEC (1)

Subject to :

$$C_{1,1} = P_{1,1} \quad (2)$$

$$C_{1,j} = C_{1,j-1} + P_{1,j}\ ,\ j = 2..m \quad (3)$$

$$C_{i,1} = C_{i-1,1} + P_{i-1,1}\ ,\ i = 2..n \quad (4)$$

$$C_{i,j} = \max(C_{i-1,j}, C_{i,j-1}) + P_{i,j}\ ,$$
$$i = 2..n,\ j = 2..m \quad (5)$$

$$Bj = \textstyle\sum_{i=1}^{n} P_{i,j}\ ,\ \forall j = 1..m \quad (6)$$

$$Tj = \max(C_{i,j})\ ,\ \forall\ i = 1..n,\ j = 1..m \quad (7)$$

$$Ij = Tj - Bj\ ,\forall j = 1..m \quad (8)$$

$$TEC = \textstyle\sum_{j=1}^{m}(Bj.Pej + Ij.Iej) \quad (9)$$

Equation (1) shows the objective function of the PFSSP problem, namely minimization of the TEC (objective function); Equation constraint (2) explains the completion time of the first job sequence on machine 1; Equation constraint (3) formulates the completion time of machines 2 to m for the first-order job; Equation constraint (4) describes the completion time of the sequence i (sequence 2 to n) jobs processed on machine 1; Equation constraint (5) shows the completion time of sequence i jobs (sequence 2 to n) on machine j (machines 2 to m); Equation constraint (6) explains the total machine busy time on each machine j; Equation constraint (7) shows the completion time of machine j from the permutation sequence; Equation constraint (8) shows the total idle time of the machine j permutation sequence, and Equation constraint (9) formulates PFSSP for energy consumption.

Algorithm Hybrid Grasshopper Algorithm Optimization

This research proposes a Hybrid Grasshopper Algorithm Optimization (HGAO) to minimize energy consumption in the PFSSP problem. HGAO is a combination of Grasshopper Algorithm Optimization and local search processes. The researchers propose transforming grasshopper positions into job permutation sequences by applying Large Rank Value (LRV). LRV is an effective way of transforming continuous values into job permutations (X. Li & Yin, 2013). Continuous values are sorted from the largest to the smallest values in the LRV. Figure 1 displays the illustration of LRV. In addition, the researchers use a local search swap and flip rules to improve the performance of Grasshopper Algorithm Optimization. The description of the swap is illustrated in Figure 2. During swap operations, two locations are chosen randomly and exchanged. Whereas, a flip is achieved by flipping the randomly selected job sequence. The swap operation is shown in Figure 3. The procedure can see the HGAO pseudocode in algorithm 1.

The Grasshopper algorithm is an optimization algorithm that can be used for decision-making. This algorithm is mathematically designed to model and simulate the behavior of grasshopper in search of food. After one of the grasshopper members finds a food source, the other grasshopper herd goes to the food source. This algorithm has been proposed by Saremi et al. (2017).

The grasshopper interaction model is modeled in equation (10). Where *Xi* is the position of the *i* grasshopper. *Si* denotes the social interaction of the *i* grasshopper. *Gi* is the gravity pressure of the *i* grasshopper, and *Ai* is the influence of the *i* grasshopper wind.

$$Xi = Si + Gi + Ai \tag{10}$$

The value of S is expressed in equation 11. d_{ij} is the distance between grasshopper i to j, $\widehat{d_{ij}}$ is the distance between grasshopper i to j, and *N* is the number of grasshoppers. The two elements are formulated in equations 12 and 13.

$$S_i = \sum_{j=1}^{N} s(d_{ij})\widehat{d_{ij}} \tag{11}$$

$$\widehat{d_{ij}} = \frac{|Xj - Xi|}{dij} \tag{12}$$

$$d_{ij} = |Xj - Xi| \tag{13}$$

Social pressure on grasshoppers can be formulated in equations (14). *f* is the force of attraction, and *l* is the duration of the scale of attraction. Then evaluate *G*i and *Ai* formulated in equations (15) and (16). *g* reflects gravity, $\widehat{e}$.g., implies vector unity concerning the center of the earth, *u* is current, $\widehat{e_w}$ is vector unity concerning the direction of the wind.

$$s(r) = f^{\frac{-r}{l}} - e^{-r} \tag{14}$$

$$Gi = -g\widehat{e_g} \tag{15}$$

$$Ai = u\,\widehat{e_w} \tag{16}$$

Equations 11 to 16, which are substituted for equation (10), becomes a formulation in equation (17).

$$Xi = \sum_{\substack{J=1 \\ J\neq i}}^{N} s(|X_j - X_i|)\frac{X_j - X_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w} \tag{17}$$

The mathematical model of equation (17) statement above cannot be applied directly to solve optimization problems because grasshoppers quickly return to their comfort zones and do not communicate with other grasshoppers. The above formulation is then transformed into an equation (18). ub_d is the upper limit of the dimension d, lb_d is the lower limit of the dimensions, $\widehat{T_d}$ is the lower limit of the dimension d, c is the reduction coefficient.

$$X_i^d = C\ \Big(\sum_{\substack{j=1 \\ j\neq i}}^{N} C\ \frac{ub_d - lb_d}{2}\ S\ (|X_j^d - X_i^d|)\ \frac{X_j - X_i}{d_{ij}}\Big) + \widehat{T_d} \qquad (18)$$

Algorithm 1. pseudocode dari Algoritma HGAO

Initialize the swarm Xi (i=1,2,…,n)
Initialize cmax, cmin, and maximum number of iteration
Apply LRV on each search agent to be mapped into a sequence
Calculate the fitness of each search agent
T= the best search agent
While (1< Max number of iterations (t))
Update c using Eq. (19)
  For each search agents
    Normalize the distance between grasshopper by the equation [1, 4]
    Update the position of the current search agent by the equation (18)
    Bring the current search agent back if it goes outside the boundaries
    end for
  Update T if there is a better solution
  for $i = 0: 0.01 \times n$
    Perform swap mutation on the search agent X^{t+1}
    if (evaluate (X^{t+1}*)* $<$ *evaluate (*T^**))*
      $T^* = X^{t+1}$
    end if
  end for
  Perform *a flip operation on a random search agent* X^{t+1}
  for $i = 0: 0.01 \times n$
  if (evaluate (X^{t+1}*)* $<$ *evaluate (*T^**))*
    $T^* = X^{t+1}$
  end if
  end for
  I = I+1
end while
Return T

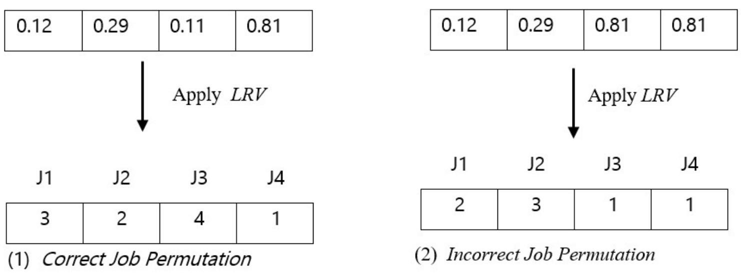


Figure 1. Illustration of LRV

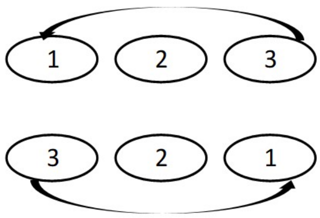


Figure 2. Do Swap illustration

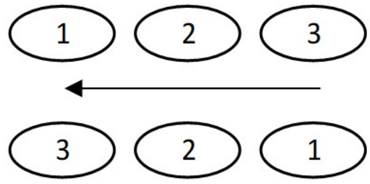


Figure 3. *Do flip illustration*

Table 1. Processing time data for each machine and job

No	*Job*	Pi,j **(minutes)**					
		M1	**M2**	**M3**	**M4**	**M5**	**M6**
1	J1	311.11	186.67	30	13.33	20	20
2	J2	27.78	0	0	33.33	0	**0**
3	J3	13.89	0	0	16.67	0	**0**
4	J4	222.22	0	0	266.67	0	**0**
5	J5	194.44	116.67	0	33.33	0	**0**
6	J6	28.11	16.87	3.3	1.47	2.2	2.2
7	J7	2809.44	1685.67	116.7	51.87	77.8	77.8
8	J8	453.44	272.07	0	19.43	0	0
9	J9	252.78	151.67	0	151.67	0	0
10	J10	695.44	0	0	834.53	0	0
11	J11	252.78	151.67	0	151.67	0	0
12	J12	27.56	0	0	33.07	0	0
13	J13	2916.67	1750	225	100	150	150

Table 2. Data on energy consumption per machine

Mesin	Pej (Kw/minutes)	lej (Kw/minutes)
1	0.0733	0.0073
2	0.0029	0
3	0.0583	0.0058
4	0.0638	0.0039
5	0.0183	0
6	0.05	0

Keep in mind that the direction of the wind is not always going to the target. There is only one position vector in the GOA algorithm. The GOA algorithm always changes to the last position. The formula used in equation (19) was used to determine the value of c. $cmax$ is the maximum value (using 1 and 0.00001), $cmin$ the minimum value (using 1 and 0.00001), l is the most recent iteration, and L is the maximum number of iterations.

$$c = cmax - l\frac{cmax-cmi}{L} \tag{19}$$

Method of collect and experiment data

The researchers perform case studies in offset printing firms. Six machines must process thirteen jobs. The details on the processing time of each machine and job can be shown in Table 1. The energy consumption of each machine is seen in Table 2. This research used a combination of parameters to assess the influence of the parameters on the objective function. The parameters used for the experiment were the population number and the number of iterations. The population uses three stages, namely 10, 50, and 100. The number of iterations is based on eight stages, namely 10, 20, 30, 40, 50, 60, 80, and 100. Therefore, there were 24 experiments to be performed.

Besides, to test the efficiency of algorithms, the HGOA algorithm was compared toward the Genetic Algorithm. The experiment was carried out with the Matlab R2014a Windows 10 Intel (R) Core (TM) i3-2348 M CPU 2 GB RAM program.

III. RESULT AND DISCUSSION

Characteristics of Respondents

The HGOA experiment findings on the combination of population and iteration show that population and iteration parameters have an impact on the quality of the solution (Table 3). The higher the iteration is used, the lower the energy consumption generated. Meanwhile, it could have an impact on the computing time that's getting longer. It also has been confirmed by several scholars, as in the study by Sugioko (2013) and Utama (D. M. Utama, T. Baroto, et al., 2019). They argue that a large number of iterations make the calculation time longer. Utama et al. (2019) explain that the larger the population, the better the solution's quality. If the iteration is getting bigger, the quality of the solution also is better. The total energy consumption based on HGOA is 819.04 KW.

Furthermore, the HGOA algorithm is compared to the Genetic Algorithm (GA). Result; studies using the Ga algorithm produce a total energy consumption of 932.69 KW (Figure 4). The total energy consumption disparity is 113,648 KW. It indicates that HGOA is successful in resolving scheduling issues. The resulting efficiency is 12.18 percent. The experimental result also reduced the idle time by 48.74 percent. It is shown that scheduling using the HGOA algorithm is more efficient and better than

Table 3. Results of the HGOA experiment on a combination of population and iteration

Population	Iteration	TEC (KW)	Job Order	Computation Time (seconds)
10	10	843.72	7-12-4-10-6-2-11-9-1-13-5-3-8	1.406
	20	820.59	13-7-4-1-2-5-8-9-10-11-12-3-6	1.422
	30	832.66	7-9-12-2-4-5-13-3-8-6-10-1-11	1.375
	40	826.81	13-10-2-7-9-12-5-11-3-4-6-1-8	1.422
	50	819.14	13-7-2-6-11-10-4-8-5-1-9-3-12	9.578
	60	819.04	13-7-6-4-12-3-8-11-9-10-1-5-2	6.875
	80	819.08	13-7-9-10-11-12-5-3-2-1-8-4-6	6.922
	100	819.06	13-7-4-10-8-9-6-11-1-2-5-12-3	6.891
50	10	819.08	13-7-2-11-12-6-5-10-1-3-9-8-4	35.188
	20	819.06	13-7-9-12-11-10-4-3-6-8-2-1-5	34.875
	30	819.04	13-7-2-5-10-3-4-1-6-8-9-11-12	35.047
	40	819.04	13-7-9-11-10-3-1-5-12-8-2-6-4	35.188
	50	819.04	13-7-12-3-6-2-10-8-9-1-11-5-4	184.063
	60	819.04	13-7-3-9-10-5-1-6-2-11-12-8-4	197.594
	80	819.04	13-7-12-3-2-10-11-1-6-9-5-8-4	175.703
	100	819.04	13-7-3-9-4-12-11-8-1-6-10-5-2	176.438
100	10	819.04	13-7-6-5-10-8-9-1-11-3-2-12-4	165.906
	20	819.04	13-7-10-1-2-3-5-6-8-9-11-12-4	162.656
	30	819.04	13-7-3-12-10-9-6-2-11-5-1-8-4	162.859
	40	819.04	13-7-11-3-10-1-6-4-8-9-12-5-2	148.266
	50	819.04	13-7-12-2-6-10-1-11-8-3-9-5-4	752.203
	60	819.04	13-7-10-8-5-11-1-6-9-3-2-12-4	762.969
	80	819.04	13-7-12-11-4-1-9-6-8-10-3-5-2	736.969
	100	819.04	13-7-4-10-2-1-8-3-6-11-9-5-12	816.938

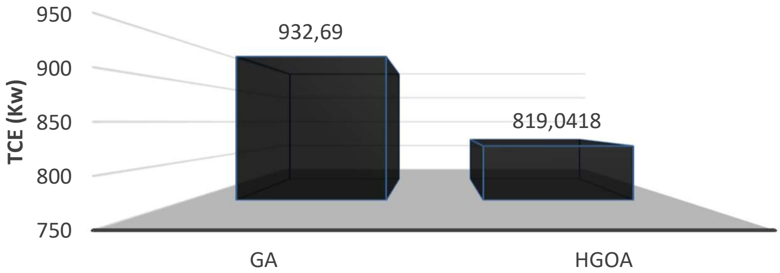
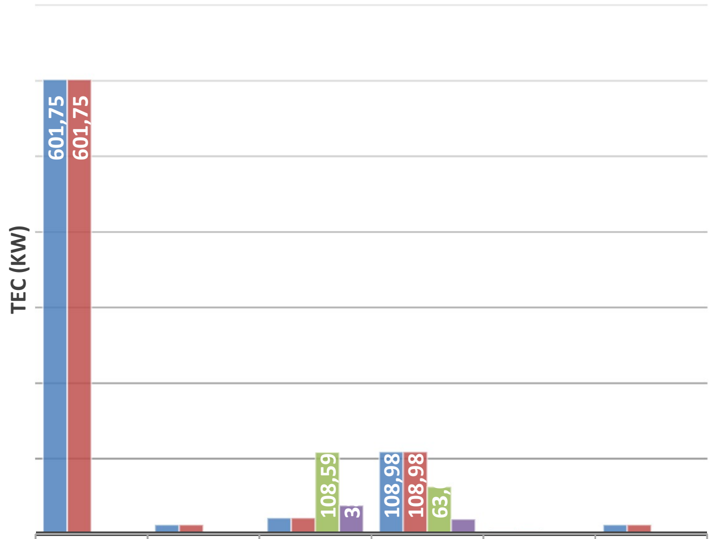


Figure 4. Comparison of HGOA and GA



	M1	M2	M3	M4	M5	M6
Energy Consumption Busy GA	601,75	12,63	21,88	108,98	3,33	12,5
Energy Consumption Busy HGOA	601,75	12,63	21,88	108,98	3,33	12,50
Energy Consumption Idle GA	0	0	108,59	63,03	0	0
Energy Consumption Idle HGOA	0	0	38,10	19,87	0	0

Machine

Energy Consumption Busy GA
Energy Consumption Busy HGOA
Energy Consumption Idle GA
Energy Consumption Idle HGOA

Figure 5. Comparison of energy consumption when the engine is busy and idle each

using the GA form.

Figure 5 shows that energy consumption at idle has a significant impact on the overall energy consumption output. With any sequence of tasks, the energy consumption spent during the process is always the same. Therefore, leave time must be minimized to minimize energy consumption.

IV. Conclusion

This research aims to reduce energy consumption in a flow shop scheduling problem using a Hybrid Grasshopper Algorithm Optimization (HGAO). The results showed that the HGAO algorithm is capable of reducing the energy consumption of offset printing firms. The HGAO algorithm is also associated with the genetic algorithm (GA). The results show that HGAO is more effective in minimizing energy consumption. The results obtained are that the higher the iteration used, the lower the energy consumption generated. Meanwhile, it could have an impact on the computing time that's getting longer. To carry out further studies, it is essential to further improve the question of energy consumption by considering disposal time and setup.

REFERENCES

Brundage, M.P., Chang, Q., Li, Y., Xiao, G., Arinez, J. (2013). Energy efficiency management of an integrated serial production line and HVAC system. *IEEE Transactions on Automation Science and Engineering, 11*(3), 789-797.

Chen, J.-f., Wang, L., Peng, Z.-p. (2019). A collaborative optimization algorithm for energy-efficient multi-objective distributed no-idle flow-shop scheduling. *Swarm and Evolutionary Computation, 50*, 100557. doi: https://doi.org/10.1016/j.swevo.2019.100557

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, 29*(5), 418-429. doi: https://doi.org/10.1016/j.rcim.2013.04.001

Elias, R., Yuan, M., Wahab, M., Patel, N. (2019). Quantifying saving and carbon emissions reduction by upgrading residential furnaces in Canada. *Journal of Cleaner Production, 211*, 1453-1462.

Fang, K., Uhan, N., Zhao, F., Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems, 30*(4), 234-240.

Garey, M. R., Johnson, D. S., Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research, 1*(2), 117-129. doi: https://doi.org/10.1287/moor.1.2.117

Gong, G., Chiong, R., Deng, Q., Han, W., Zhang, L., Lin, W., Li, K. (2020). Energy-efficient flexible flow shop scheduling with worker flexibility. *Expert Systems with Applications, 141*, 112902.

Grobler, J., Engelbrecht, A. P., Kok, S., Yadavalli, S. (2010). Metaheuristics for the multi-objective FJSP with sequence-dependent set-up times, auxiliary resources and machine downtime. *Annals of Operations Research, 180*(1), 165-196.

Jiang, T., Zhang, C., Zhu, H., Gu, J., Deng, G. (2018). Energy-efficient scheduling for a job shop using an improved whale optimization algorithm. *Mathematics, 6* (11), 220.

Lei, D., Gao, L., Zheng, Y. (2018). A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop. *IEEE Transactions on Engineering Management, 65*(2), 330-340. doi: https://doi.org/10.1109/TEM.2017.2774281

Li, S., Liu, F., Zhou, X. (2018). Multi-objective energy-saving scheduling for a permutation flow line. Proceedings of the Institution of Mechanical Engineers, Part B: *Journal of Engineering Manufacture, 232*(5), 879-888.

Li, X., Yin, M. (2013). An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure. *Advances in Engineering Software, 55*, 10-31. doi: https://doi.org/10.1016/j.advengsoft.2012.09.003

Liu, X., Wang, L., Kong, L., Li, F., Li, J. (2019). A Hybrid Genetic Algorithm for Minimizing Energy Consumption in Flow Shops Considering Ultra-low Idle State. *Procedia CIRP, 80*, 192-196. doi: https://doi.org/10.1016/j.procir.2018.12.013

Liu, Y., Dong, H., Lohse, N., Petrovic, S. (2016). A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. *International Journal of Production Economics, 179*, 259-272. doi: https://doi.org/10.1016/j.ijpe.2016.06.019

Lu, C., Gao, L., Li, X., Pan, Q., Wang, Q. (2017). Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *Journal of Cleaner Production, 144*, 228-238. doi: https://doi.org/10.1016/j.jclepro.2017.01.011

Maulana, S.K.D.B., Utama, D.M., Asrofi, M.S., Ningrum, I.S., Alba, N., Ahfa, H.A., Zein, T.A. (2019). The Capacitated Sustainable EOQ Models: Models Considering Tax Emissions. *Jurnal Teknik Industri, 21*(1), 12-21.

Öztop, H., Tasgetiren, M. F., Eliiyi, D. T., Pan, Q.-K., & Kandiller, L. (2020). An energy-efficient permutation flowshop scheduling problem. *Expert Systems with Applications, 150*, 113279.

Pan, Q.-K., Gao, L., Li, X.-Y., Gao, K.-Z. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Applied Mathematics and Computation, 303*, 89-112.

Piroozfard, H., Wong, K.Y., Wong, W.P. (2018). 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, 128*, 267-283.

Saremi, S., Mirjalili, S., Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30-47.

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production, 67*, 197-207.

Sugioko, A. (2013). Perbandingan Algoritma Bee Colony dengan Algoritma Bee Colony Tabu List dalam Penjadwalan Flow Shop. *Jurnal Metris, 14* (2), 113-120.

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, 81*, 82-95. doi: https://doi.org/10.1016/j.compind.2015.10.001

Tian, Y., Xiong, S., Ma, X., Ji, J. (2018). Structural path decomposition of carbon emission: A study of China's manufacturing industry. *Journal of Cleaner Production, 193*, 563-574.

Utama, D., Widodo, D., Wicaksono, W., & Ardiansyah, L. (2019). A new hybrid metaheuristics algorithm for minimizing energy consumption in the flow shop scheduling problem. *International Journal of Technology, 10*, 320-331.

Utama, D.M. (2018a). Algoritma LPT-Branch and Bound Pada Penjadwalan Flexible Flowshop untuk Meminimasi Makespan. *PROZIMA* (*Productivity, Optimization and Manufacturing System Engineering*), *2*(1), 20-26. doi: http://ojs.umsida.ac.id/index.php/prozima/article/view/1527

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

Utama, D.M. (2019a). 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: http://dx.doi.org/10.22219/JTIUMM.Vol20.No1.62-72

Utama, D.M. (2019b). *Salp Swarm Algorithm Untuk Meminimasi Konsumsi Energi Pada Penjadwalan Flow Shop Dengan Set Up dan Removal Time*. Paper presented at the Prosiding SENTRA (Seminar Teknologi dan Rekayasa).

Utama, D.M., Ardiansyah, L.R., Garside, A.K. (2019). Penjadwalan Flow Shop untuk Meminimasi Total Tardiness Menggunakan Algoritma Cross Entropy–Algoritma Genetika. *Jurnal Optimasi Sistem Industri, 18*(2), 133-141. doi: https://doi.org/10.25077/josi.v18.n2.p133-141.2019

Utama, D.M., Baroto, T., Maharani, D., Jannah, F.R., Octaria, R.A. (2019). Algoritma ant-lion optimizer untuk meminimasi emisi karbon pada penjadwalan flow shop dependent sequence set-up. *Jurnal Litbang Industri, 9*(1), 69-78.

Utama, D.M., Garside, A.K., Wicaksono, W. (2019). Pengembangan algoritma Hybrid Flow shop Three-Stage Dengan Mempertimbangkan Waktu Setup. *Jurnal Ilmiah Teknik Industri, 18* (1), 72-78. doi: http://dx.doi.org/10.23917/jiti.v18i1.7683

Utama, D.M., Widodo, D.S., Ibrahim, M.F., Dewi, S.K. (2020). An effective hybrid ant lion algorithm to minimize mean tardiness on permutation flow shop scheduling problem. *International Journal of Advances in Intelligent Informatics, 6* (1), 23-35.

Widodo, D.S., Santoso, P.B., Siswanto, E. (2014). Pendekatan Algoritma Cross Entropy-Genetic Algorithm Untuk Menurunkan Makespan Pada Penjadwalan Flow Shop. *Journal of Engineering and Management in Industrial System, 2*(1).

Widodo, D.S., Utama, D.M. (2019). Analisis Model Sustainable Economic Order Quantity Dengan Mempertimbangkan Emisi Karbon Dan Batasan Kapasitas Gudang Untuk Menekan Total Biaya Persediaan. *Teknik, 40*(3), 169-175.

Wu, X., Che, A. (2019). A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. *Omega, 82*, 155-165. doi: https://doi.org/10.1016/j.omega.2018.01.001

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, 149*, 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, 63*(1), 37-40. doi: https://doi.org/10.1016/j.cirp.2014.03.011