Hybrid Grey Wolf Algorithm for Energy-Efficient Scheduling with Sequence-Dependent Setup Times: A Case Study

D M Utama

1Department Industrial Engineering, University of Muhammadiyah Malang, Jl. Tlogomas No. 246, 65144 Malang, East Java, Indonesia
*dana@umm.ac.id

Abstract. Non-renewable energy consumption is one of the dominant factors in global warming. The industrial sector has a significant contribution to this problem. At present, the company is required to carry out efficiency, especially energy consumption, because the industry contributes to the most significant energy consumption. One effort to minimize energy consumption in the industrial sector is with proper scheduling. This research attempts to develop the Hybrid Grey Wolf Optimizer (HGWO) Algorithm to complete Energy-Efficient Scheduling (EES) on the Permutation Flow Shop Scheduling Problem (PFSP). This study considers Sequence-Dependent Setup Times on the PFSP problem. A case study was used to resolve EES on PFSP problems. The HGWO parameter experiment was also used to test the parameters in the case study solving. This research also compares HGWO with several popular procedures. The comparison of algorithms shows that the results of the HGWO algorithm are more competitive for completing EES in PFSP problems.

Keywords: energy consumption, flow shop, grey wolf, energy-efficient

1. Introduction

Recently, climate change and global warming have become the world's attention [1]. It is caused by increased non-renewable energy consumption, and it also has an impact on increasing carbon emissions in the environment [2]. Increased carbon emissions cause global warming, and an estimated half of total world energy consumption is consumed from the industrial sector. Manufacturing companies must have a role to reduce global warming [3]. Therefore, companies must pay attention to energy consumption in production operations [4]. Vuletić, et al. [5] revealed that many companies use machines that require significant energy consumption. In addition, when the machine is idle, the energy consumption required is significant [6]. However, in some PFSP problems, the idle state does not require energy consumption. Through proper scheduling, the company can reduce electricity and non-renewable energy consumption. This problem is commonly called Energy-Efficient Scheduling (EES) [7].

This EES problem focuses on Permutation Flow Shop Scheduling Problem (FSSP). This problem is a job scheduling problem that is processed via several machines in the same sequence [8] [9]. FSSP is categorized as Non-Polynomial Hard Problem [10, 11]. Based on previous research, there have been several algorithms used to solve EES on the PFSP problem. These algorithms include Nahwas Encore Ham (NEH) Algorithm [12], Ant Colony Optimization Algorithm (ACO) [13], Genetic Algorithm (GA) [14], and Campbell Dudeck Smith (CDS) [15]. Several other sophisticated algorithms have also
been proposed by researchers such as Hybrid Grasshopper Algorithm Optimization [16], Hybrid Whale Optimization Algorithm [17], Salp Swarm Algorithm [18], and Hybrid Metaheuristic [10]. Previous research generally discusses the PFSP problem with the assumption that removal time and setup are included with the processing time. Although some research has been done to resolve EES on PFSP problems, very little concern has been paid to the issue of sequence-dependent setup times. Based on the description above, several algorithms have been proposed to solve EES on PFSP problems. However, no studies are utilizing the Grey Wolf Optimizer (GWO) procedure to complete the PSFP problem. GWO is a novel algorithm that is inspired by grey wolf behavior in hunting for victims [19]. It has been effectively applied to the completion of two-stage assembly flow shop scheduling [20] and dynamic scheduling [21]. These motivate the author to use GWO to solve EES on PFSP problems. Therefore, this study aims to propose a Hybrid Grey Wolf Optimizer (HGWO) Algorithm to resolve EES on PFSP problems. This research takes the form of a case study of PFSP scheduling problems in convection companies. It is hoped that this research will contribute to a deeper understanding of EES problems in PFSP with sequence-dependent setup times.

2. Methods

2.1 Assumptions, notations and problem definitions

The PFSP problem assumes (1) a number of n jobs (n=1, 2, 3 ... i) in the same order worked on a series of m machines (m=1, 2, 3 ... j), (2) the time of the process of the i job on the jth machine notated as Pij, (3) all machines available at t = 0, (4) setup time for first-order jobs are independent of the job sequence (5) the time of setup for job transfers from job i-1 to the job i notated as Si−1,i, (6) the time of removal of each job is separate from the time of the process. (7) each machine that stops independently of the other machines (each machine stops when the last job on each machine finishes). (8) each job when it starts will be processed until it finishes (cannot be interrupted). The notations used in this problem are conveyed as follows.

\n\begin{align*}
  i & : \text{index of jobs, } i = 1, 2, \ldots, n \\
  j & : \text{index of machines, } j = 1, 2, \ldots, m \\
  m & : \text{number of machines} \\
  n & : \text{number of jobs} \\
  P_i, j & : \text{time of the process of job } i \text{ on machines } j \\
  S_{i-1, i} & : \text{time of setup move sequence } i-1 \text{ to } i \text{ on machine} \\
  S_i & : \text{time of setup of job } i \text{ in the first sequence on each machine} \\
  R_i, j & : \text{time of removal for job } i \text{ on machine } j \\
  Pe_j & : \text{consumption of energy on machine } j \\
  Re_j & : \text{consumption of energy on machine } j \text{ when removal} \\
  Ie_j & : \text{consumption of energy on machine } j \text{ when idle} \\
  Se_j & : \text{consumption of energy Setup on machine } j \\
  ECI & : \text{total consumption of energy in idle} \\
  ECS & : \text{total consumption of energy in setup} \\
  ECR & : \text{total consumption of energy in removal} \\
  ECP & : \text{total consumption of energy in the process} \\
  Ci, j & : \text{finish time on job sequence } l \text{ at on machines } j \\
  T_j & : \text{finish time on machines } j \\
  B_j & : \text{time of total busy on machines } j \\
  I_j & : \text{time of total idle on machines } j \\
  S_j & : \text{time of total setup on machines } j
\end{align*}

2
\[R_j\]: time of removal on machines \(j\)

\[TEC\]: total energy consumption

\[Pop\]: Population

\[It\]: Iteration

The sequence-dependent setup times PSFP model problem has been developed from the model proposed by Li, et al. [22]. The PFSP model with sequence-dependent setup times to reduce energy consumption is as follows:

The Objective function \(Z = \min TEC\) 

Constraint :

\[C_{1,1} = S_1 + P_{1,1} + R_{1,1}\] 

\[C_{i,j} = \max(C_{i,j-1} - R_{i,j-1}, S_j) + P_{i,j} + R_{i,j} \quad j = 2 \ldots m\]

\[C_{i,1} = C_{i-1,1} + S_{i-1,1} + P_{i,1} + R_{i,1}, \quad i = 2 \ldots n\]

\[C_{i,j} = \max(C_{i,j-1} - R_{i,j-1}, S_{i-1,j} + C_{i-1,j}) + P_{i,j} + R_{i,j}, \quad i = 2 \ldots n, j = 2 \ldots m\]

\[B_j = \sum_{i=1}^{n} P_{i,j}, \quad \forall j = 1 \ldots m\]

\[S_j = \sum_{i=2}^{n} S_{i-1,i} + S_j, \quad \forall j = 1 \ldots m\]

\[R_j = \sum_{i=1}^{n} R_{i,j}, \quad \forall j = 1 \ldots m\]

\[T_j = \max(C_{i,j}), \quad \forall i = 1 \ldots n, j = 1 \ldots m\]

\[I_j = T_j - B_j - S_j - R_j, \quad \forall j = 1 \ldots m\]

\[TEC = \sum_{j=1}^{m}(B_j, P_e j + I_j, L e j + S_j, S e j + R_j, R e j)\]

Equation (1) illustrates the function to minimize TEC; Constraint (2) explains the finish time of the 1st job sequence in the machine 1; Constraint (3) describes the finish time of job order 1 in the machines 2 to \(m\); Constraint (4) explains the finish time of job order \(i\) in the machine 1; Constraint (5) shows the job finish time in job order \(i\) in the machine \(j\); Constraint (6) explains the total time of busy in the machine \(j\); Constraint (7) describes the total setup time in the machine \(j\). Constraint (8) describes the total time of removal in the machine \(j\). Constraint (9) describes the completion time of machine \(j\) from permutation; Constraint (10) describes the total time of idle in the machine \(j\); Constraint (11) formulates total energy consumption.

2.2 Proposed Hybrid Grey Wolf Optimizer (HGWO) Algorithm

Hybrid Grey Wolf Optimizer (HGWO) is proposed to solve EES on PFSP problems. Minimizing total energy consumption is the objective function in this problem. The author proposes HGWO that inspired by the grey wolf’s behavior in hunting for prey combined with local search procedures. The Large Rank Value (LRV) procedure is proposed to transform GWO positions into job sequences. LRV is a potent procedure to transform continuous values into job sequence (job permutations) [23]. In this procedure, the GWO position is arranged by the largest value to the smallest value. Figure 1 is an illustration of the conversion of the GWO position (continuous value) to a permutation job. Local search swap and flip rules are proposed to improve the GWO solution. Figure 2 is a swap illustration proposed to swap two randomly selected positions. Furthermore, Figure 3 is an illustration of Flip’s rule, which is done by reversing the randomly selected job sequence. The swap and the flip procedure is repeated as much as 0.01*n. Pseudocode The proposed HGWO algorithm is presented in algorithm 1.
The GWO algorithm is inspired by the grey wolf hunting in nature. Grey wolves are considered to be at the top of the food chain. They prefer to exist in the entourage, with an average of 5 - 12 individuals. Within this group, wolves are classified into four categories of roles, namely alpha (α), beta (β), delta (δ), and omega (ω). Based on its behavior, there are three hunting phases: tracking, siege, and attack used for optimization. From this behavior, the siege was formulated with equations (12), (13), (14), and (15).

\[
D = |\dot{C} \cdot \ddot{X}_p(t) - \ddot{X}(t)|
\]

\[
\ddot{X}(t + 1) = \ddot{X}_p(t) - \dot{A} \cdot D
\]

\[
\dot{A} = 2\dddot{a} \cdot \dddot{r}_1 - \dddot{a}
\]

\[
\dddot{C} = 2 \cdot \dddot{r}_2
\]

Where \( t \) states iteration (iteration). The prey position vector is notated by \( \dddot{X}_p \), and the grey wolf position vector is described by \( \dddot{X} \). \( \dddot{D} \), \( \dddot{A} \), and \( \dddot{C} \) represent the vector coefficients. For mathematical modeling in attacking prey, it is indicated by the value of \( \dddot{a} \) between values 2 to 0 for iteration while \( \dddot{r}_1 \) and \( \dddot{r}_2 \) are random vectors with values from 0 to 1.

Hunting is led by α, while β and δ sometimes hunt. Therefore based on the social hierarchy, α is the candidate for the first, β second, and δ third-best solutions. In finding the optimal position, hunting is represented by equations (16), (17) and (18) as follows:

\[
\dddot{D}_\alpha = |\dddot{C}_1 \cdot \dddot{X}_\alpha - \dddot{X}|, \dddot{D}_\beta = |\dddot{C}_2 \cdot \dddot{X}_\beta - \dddot{X}|, \dddot{D}_\delta = |\dddot{C}_3 \cdot \dddot{X}_\delta - \dddot{X}|
\]

\[
\dddot{X}_1 = \dddot{X}_\alpha - \dddot{A}_1 \cdot (\dddot{D}_\alpha), \dddot{X}_2 = \dddot{X}_\beta - \dddot{A}_2 \cdot (\dddot{D}_\beta), \dddot{X}_3 = \dddot{X}_\delta - \dddot{A}_3 \cdot (\dddot{D}_\delta)
\]

\[
\dddot{X}(t + 1) = \frac{\dddot{X}_1 + \dddot{X}_2 + \dddot{X}_3}{3}
\]
Algorithm 1. Pseudocode of proposed HGWO Algorithm

```
Initialization of the grey wolf position Xi (i=1, 2, 3,...,n)
Convert position based on LRV Procedure in every grey wolf
Initialization value of a, A and C
compute the fitness of each search agent utilizing equation 1 to 11
The best search agent is notated Xa
The second best search agent is notated Xβ
The third best search agent is notated Xδ
While(t<Max number of iterations)
    for each search agent
        Regenerate the current search agent position of GWO utilizing equation (18)
    End for
    Update a, A and C
    Convert position based on LRV Procedure in every grey wolf
    Compute the fitness of all search agent GWO using equation 1 to 11
    Update the Xa, Xβ, Xδ
    for i = 0: 0.01 × n
        Perform the swap mutation on the search agent X^{t+1}
        if (evaluate (X^{t+1}) < evaluate (Xa))
            Xa = X^{t+1}
        end if
    end for
    for i = 0: 0.01 × n
        Perform the flip operation on a random search agent X^{t+1}
        if (evaluate (X^{t+1}) < evaluate (Xa))
            Xa = X^{t+1}
        end if
    end for
    t= t+1
end while
return Xa
```

2.3 Data and Experimental Settings

A case study was conducted at a convection company. Data on processing time and removal time can be seen in table 1. Data on energy requirements and setup time for each machine in the first sequence job are presented in table 2. Setup-time-dependent data are presented in table 3. In this case study, ten jobs must be completed in 3 machines in sequence. The study used iteration and population parameters with a value of 100, 200, and 500 each to test the parameters of the HGWO algorithm. There were nine combinations of parameter experiments used, and each combination of experiments was carried out ten times. In total, there were 90 experimental combinations of HGWO algorithm parameters. The parameter combination trial was used to examine the influence of the HGWO parameter on the completion of the EES on PFSP problem.

Each parameter combination experiment was recorded the total energy consumption and computational time needed to solve the problem. Computation time needs to be recorded to determine the efficiency of the parameters used in the completion of EES on PFSP. The experiment was carried out using Matlab R2018a software on Windows 10 Intel (R) Core (TM) i3-2348M CPU RAM 3 Gb. Parameter experiment analysis was carried out using Box-plot Analysis assisted with SPSS software. This study compared the HGWO algorithm with several popular algorithms such as GA [14], NEH [12], and CDS [15] to examine the performance of the algorithm of HGWO. In this comparison, the
parameter used in the HGWO algorithm was a population of 500 with 200 iterations. In the GA algorithm, this study used 500 chromosomes and 200 iterations. Each energy consumption structure in each algorithm was recorded, and it was compared.

### Table 1. Processing time and removal time (minute)

| Job | Processing Time | Time of Removal |
|-----|-----------------|-----------------|
|     | M1 | M2 | M3 | M1 | M2 | M3 |
| 1   | 10.4| 33 | 8.2| 0  | 2  | 0  |
| 2   | 4   | 15 | 3.6| 0  | 1  | 0  |
| 3   | 20  | 33 | 17.9| 0  | 3  | 0  |
| 4   | 4   | 13 | 3.6| 0  | 2  | 0  |
| 5   | 6.2 | 30 | 7.2| 0  | 3  | 0  |
| 6   | 3.1 | 27 | 3.1| 0  | 1  | 0  |
| 7   | 9.3 | 13 | 10.8| 0  | 3  | 0  |
| 8   | 10.3| 26 | 10.2| 0  | 2  | 0  |
| 9   | 8.2 | 34 | 14.3| 0  | 3  | 0  |
| 10  | 8   | 26 | 16.2| 0  | 2  | 0  |

### Table 2. Data on energy requirements and setup time for each machine in the first sequence job

| Machine | $S_{ej}$ | $P_{ej}$ | $R_{ej}$ | $I_{ej}$ | $S_i$ |
|---------|----------|----------|----------|----------|-------|
| M1      | 0.3      | 0.5      | 0        | 0        | 1     |
| M2      | 0.7      | 1.5      | 0.5      | 0        | 5     |
| M3      | 0.2      | 0.5      | 0        | 0        | 1     |

### Table 3. Setup time to move sequence ($S_{i-1,i}$) in minute

| $S_{i-1,i}$ | Job i |
|-------------|-------|
| 1           | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 12    | 12    | 12    | 12    | 12    | 12    | 24    |
| 2           | 12    | 0     | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 18    | 24    |
| 3           | 12    | 12    | 0     | 6     | 6     | 6     | 6     | 6     | 6     | 6     | 6     | 6     | 12    | 12    | 18    | 30    | 18    | 30    | 18    | 30    | 18    | 18    | 18    | 24    |
| 4           | 12    | 12    | 12    | 0     | 12    | 6     | 6     | 6     | 6     | 6     | 6     | 6     | 12    | 12    | 18    | 30    | 18    | 30    | 18    | 30    | 18    | 18    | 18    | 24    |
| 5           | 6     | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 14    | 18    | 24    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 18    | 24    |
| i-1         | 6     | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 12    | 18    | 24    |

### 3. Results and Discussion

The outcome of the HGWO parameter trial on total energy consumption can be seen in Figure 4. It shows that the larger the population and the iteration of HGWO used, the smaller the energy consumption produced. Conversely, the lower the population and HGWO iteration, the higher the consumption of energy produced and varied. These results correspond with the study conducted by
Utama, et al. [16]. Interestingly, in a population of 500 iterations of 200 and 500, the energy consumption produced is the same (1366.4 watts).

Figure 5 shows the comparison of computational time (second) on the HGWO parameter experiment. It results illustrate that the higher the population and HGWO iteration used, the higher the computational time required. Conversely, the smaller the population and HGWO iteration, the computational time required is low. These results also correspond with an investigation carried out by Utama, et al. [16]. The most striking result that emerged from the data was that in a population of 500 iterations of 200 and 500, the energy consumption produced was the same (1366.4 watts). However, the computational time required in a population of 500 and 500 iterations is relatively large. Therefore, in the case of 10 jobs, we prefer 500 population parameters and 200 iterations.

![Figure 4. Comparison of energy consumption (watts) in the HGWO parameter experiment](image)

Comparison of Algorithms and energy structures can be seen in Figure 5. It shows that the total energy consumption of the HGWO algorithm is better and more efficient compared to several other algorithms, such as GA, NEH, and CDS. Most interestingly, the total energy consumption on this issue is influenced by the consumption of energy setup (see Figure 6). Energy consumption of process, removal, and idle does not affect total energy consumption. Therefore, for the problem of sequence-dependent setup time, setup time needs to be minimized to minimize energy consumption.
**Figure 5.** Comparison of computational time (second) on HGWO parameter experiments

**Figure 6.** Comparison of Algorithms and energy structures

4. **Conclusion**

The research successfully developed the Hybrid Grey Wolf Optimizer (HGWO) Algorithm to solve EES on PFSP problems. The results showed that the larger the population and HGWO iteration used,
the smaller the consumption of energy produced. In addition, the results of comparisons with other algorithms show that the HGWO algorithm produces competitive solutions compared to different algorithms. This study has limitations because it only uses a small number of jobs (10 jobs). Therefore, for further research, we suggest investigating with medium and large numbers of jobs.

5. References
[1] D. M. Utama, 2019  An effective hybrid sine cosine algorithm to minimize carbon emission on flow-shop scheduling sequence dependent setup  Jurnal Teknik Industri 20 62-72
[2] X. Wu and A. Che, 2019  A memetic differential evolution algorithm for energy-efficient parallel machine scheduling  Omega 82 155-165
[3] K. Fang, N. Uhan, F. Zhao, and J. W. Sutherland, 2011  A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction  Journal of Manufacturing Systems 30 234-240
[4] H. Zhang, F. Zhao, K. Fang, and J. W. Sutherland, 2014  Energy-conscious flow shop scheduling under time-of-use electricity tariffs  CIRP Annals-Manufacturing Technology 63 37-40
[5] D. Vuletić, M. Stojanovska, M. Avdibegović, R. Nevenić, N. Petrović, S. Posavec, et al., 2010 Forest-related conflicts in the south-east European region: regional aspects and case studies in Albania, Bosnia and Herzegovina, Croatia, Macedonia and Serbia  in Support of Good Governance 117
[6] G. Mouzon, M. B. Yildirim, and J. Twomey, 2007 Operational methods for minimization of energy consumption of manufacturing equipment  International Journal of Production Research 45 4247-4271
[7] C. Gahm, F. Denz, M. Dirr, and A. Tuma, 2016 Energy-efficient scheduling in manufacturing companies: A review and research framework  European Journal of Operational Research 248 744-757
[8] D. S. Widodo and D. M. Utama, 2020 The Hybrid Ant Lion Optimization Flow Shop Scheduling Problem for Minimizing Completion Time  Journal of Physics: Conference Series 1569 022097
[9] D. M. Utama, D. S. Widodo, M. F. Ibrahim, and S. K. Dewi, 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 23-35
[10] D. Utama, D. Widodo, W. Wicaksono, and L. Ardiansyah, 2019 A new hybrid metaheuristics algorithm for minimizing energy consumption in the flow shop scheduling problem  International Journal of Technology 10 320-331
[11] M. R. Garey, D. S. Johnson, and R. Sethi, 1976 The Complexity of Flowshop and Jobshop Scheduling  Mathematics of Operations Research 1 117-129
[12] G.-S. Liu, J.-J. Li, and Y.-S. Tang, 2018 Minimizing total idle energy consumption in the permutation flow shop scheduling problem  Asia-Pacific Journal of Operational Research 35 1850041
[13] L. Tuo, L. Dai, and X. Chen, 2014 Scheduling of Discrete Manufacturing Process for Energy Saving  Applied Mechanics and Materials 556 4248-4254
[14] J. Luo, S. Fujimura, D. El Baz, and B. Plazolles, 2019 GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem  Journal of Parallel and Distributed Computing 133 244-257
[15] D. M. Utama, 2019 Pengembangan Algoritma NEH Dan CDS Untuk Meminimasi Consumption Energy Pada Penjadwalan Flow Shop  Prosiding SENTRA (Seminar Teknologi dan Rekayasa) 47-54
[16] D. M. Utama, T. Baroto, and D. S. Widodo, 2020 Energy-Efficient Flow Shop Scheduling Using Hybrid Grasshopper Algorithm Optimization  Jurnal Ilmiah Teknik Industri 19 30-38
[17] D. M. Utama, D. S. Widodo, M. F. Ibrahim, K. Hidayat, T. Baroto, and A. Yurifah, 2020 The hybrid whale optimization algorithm: A new metaheuristic algorithm for energy-efficient on flow shop with dependent sequence setup *Journal of Physics: Conference Series* **1569** 022094

[18] D. M. Utama, 2019 Salp Swarm Algorithm Untuk Meminimasi Konsumsi Energi Pada Penjadwalan Flow Shop Dengan Set Up Dan Removal Time *Prosiding SENTRA (Seminar Teknologi dan Rekayasa)* 79-85

[19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, 2014 Grey wolf optimizer *Advances in engineering software* **69** 46-61

[20] G. Komaki and V. Kayvanfar, 2015 Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time *Journal of Computational Science* **8** 109-120

[21] C. Lu, L. Gao, X. Li, and S. Xiao, 2017 A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry *Engineering Applications of Artificial Intelligence* **57** 61-79

[22] S. Li, F. Liu, and X. Zhou, 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** 879-888

[23] X. Li and M. Yin, 2013 An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure *Advances in Engineering Software* **55** 10-31