The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2022

Volume 21, Pages 441-451

IConTES 2022: International Conference on Technology, Engineering and Science

Whale Optimization Algorithms for Multi-Objective Flowshop Scheduling Problems

Cecilia E. NUGRAHENI
Parahyangan Catholic University

Luciana ABEDNEGO
Parahyangan Catholic University

Craven S. SAPUTRA
Parahyangan Catholic University

Abstract: One of the most common scheduling classes in the industry is Flow Shop Scheduling Problem (FSP). Given a set of jobs that must be completed in a series of identical stages, each stage is performed by a single machine. The goal of the FSP is to find a series of jobs that meets specific optimization criteria. Multi-objective FSP (MOFSP) is an FSP with more than one optimization target. This study investigates the MOFSP solution using two variants of the Whale Optimization Algorithm technique, namely the basic Whale Optimization Algorithm (WOA) and an improved Whale Optimization Algorithm (I-WOA). The objective criteria are makespan and total flow time. This study aims to examine the performance of WOA and I-WOA and determine how the weight ratio of optimization criteria affects each method. Several experiments were conducted using the Taillard Benchmark, and it was concluded that in general, WOA outperforms I-WOA, and the best weight ratio is makespan:total flow time is 75:25.

Keywords: Scheduling, Flow shop, Multi-objective FSP, Whale optimization algorithm

Introduction

One class of scheduling problems commonly found in the industry is the Flow Shop Scheduling Problem (FSP). Given a set of jobs that must be processed in a series of stages with only one machine for processing the jobs at each stage, FSP aims to find a sequence of jobs that meets specific optimal criteria or objectives. FSP can therefore be viewed as an optimization problem. FSP with two or more objectives is called Multi-objective FSP (MOFSP) (Yenisey & Yagmahan, 2014).

The production process frequently uses a variety of optimization goals. Makespan and total flow time are two of them. Makespan is the total amount of time needed to complete all jobs, starting with the first job on the first machine and ending with the last job on the last machine. Flow time is the time needed to finish a job, i.e., the duration from a job is ready to be processed by the first machine and completed by the last machine. Total flow time is the sum of all the jobs' flow times. In this study, we are interested in solving MOFSP with the goal is to minimize the makespan and total flow time.

Scheduling problems, including production process scheduling problems in the manufacturing industry, are very challenging problems. Many techniques or approaches have been proposed to solve the production process scheduling problems. These approaches can be divided into two groups: the exact approach and the heuristic approach. Although the heuristic approach does not guarantee an optimal solution, it can provide a near-optimal solution in an acceptable time. Moreover, heuristic techniques are generally divided into two types, namely

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

constructive and improvement heuristics (Nugraheni & Abednego, 2016a). Some of the algorithms included in the constructive technique are dispatching rules such as FIFO, LIFO, SPT, LPT, and some popular algorithms such as NEH, CDS, Palmer, Gupta, and Pour. Improvement heuristics are called metaheuristics, such as genetic algorithm, simulated annealing, particle swarm optimization, firefly algorithm, bee colony algorithm, and whale optimization algorithm. Metaheuristics are widely used to solve production scheduling problems. Many studies have been conducted using metaheuristics to scheduling problems with multiple objectives (Demir & Gelen, 2021; Lu, Xiao, Li, & Gao, 2016; Singh, Oberoi & Singh, 2020; Schulz, Buscher & Shen, L., 2020; Yenisey & Yagmahan, 2014).

Whale optimization Algorithm (WOA) is a swarm intelligence optimization algorithm proposed by Seyedali Mirjalili and Andrew Lewis in 2016 (Alimoradi, 2021; Mirjalili & Lewis, 2016; Rana et al., 2020). WOA simulates mimics the hunting behavior of humpback whales called bubble-net feeding method (Mirjalili & Lewis, 2016), as shown in Figure 1. The whale will create distinctive bubbles along a circle or a spiral path once it finds its prey. WOA has been widely used to solve optimization problems, including production scheduling problems.

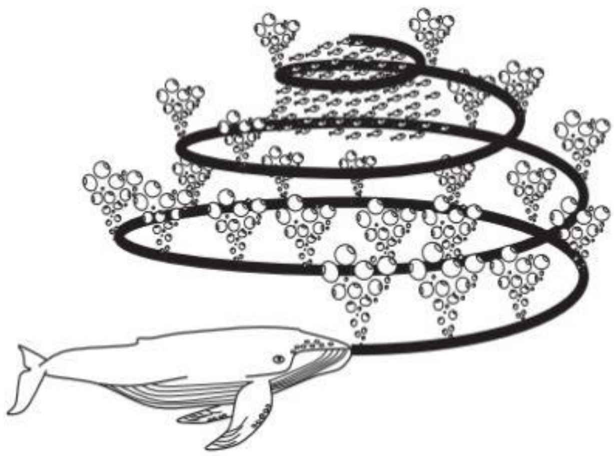

Figure 1. Bubble-net feeding behavior of humpback whales. (Mirjalili & Lewis, 2016)

There are many variations of the standard WOA from Mirjalili & Andrews proposed by the researchers, including *improved* WOA (Ning, & Cao, 2021; Wang, Deng, Zhu, & Hu, 2019), *boosted* WOA *enhanced* WOA (Cao, Xu, Yang, Dong, & Li, 2022; Chakraborty, Saha, Sharma, Mirjalili, & Chakraborty, 2020; Hassouneh et.al, 2021; Nadimi-Shahraki, Zamani, & Mirjalili, 2022), *augmented* WOA (Alnowibet, Shekhawat, Saxena, Sallam, & Mohamed, 2022), *modified* WOA (Liang, Xu, Siwen, Liu, & Sun, 2022) and *hybrid* WOA (Lin, Wu, Huang, & Li, 2018).

In this study, two types of Whale Optimization Algorithm, namely the basic (WOA) and the improved proposed by Wang et.al (I-WOA), will be used to solve FSP. Using a benchmark from Taillard, we conducted some experiments to compare the algorithms' performance. The rest of the paper is structured as follows. Section 2 describes the method, including WOA, I-WOA, and the implementation (computer program) of WOA and I-WOA developed in this study. Section 3 explains the computational experiments conducted in this study, including parameter setting and experiment results, as well as the analysis of the results. Section 4 summarizes this study's main findings and suggests future research directions.

Method

Basic Whale Optimization Algorithm (WOA)

The basic WOA consists of three main phases: prey encircling, exploitation phase through bubble-net and exploration phase, i.e., prey search. In this section, we will briefly describe the three parts.

1) Prey Encircling: Humpback whales choose their target prey through the capacity to find the location of prey. The best search agent is followed by other search agents to update their positions. This behavior is represented in Eq. 1 and Eq. 2:

$$D = |C \cdot X^*(t) - X(t)| \quad \text{Eq. 1}$$

$$X(t+1) = X^*(t) - A \cdot D \quad \text{Eq. 2}$$

where X^* denotes the position vector of the best-obtained solution, X is the position vector, t is the current iteration, |…| denotes the absolute value and · denotes the element-to-element multiplication. The coefficients A and C can be calculated as follows

$$A = 2a.r - a \quad \text{Eq. 3}$$

$$C = 2r \quad \text{Eq. 4}$$

where $r \in [0, 1]$ and a linearly decreases with every iteration from 2 to 0 as formulated as follows:

$$a = 2 - t.2/t_{\max} \quad \text{Eq. 5}$$

where $t_{\max}$ is the total iteration number.

The present position of search agents was moved closer to the ideal position by modifying the values of the vectors A and C. This procedure of updating positions in the neighborhood direction also helps in n-dimensionally encircling the prey.

2) *Exploitation phase* - bubble net attack: There are two approaches to model the bubble-net behavior, i.e., shrinking encircling mechanism and spiral update position. The shrinking encircling mechanism is achieved by gradually decreasing the value of a in Eq. 5 from 2 to 0 throughout several iterations. By choosing a random value A from the range [−1, 1], we can define the new position of a search agent anywhere between the position of the current best agent and the initial position of the agent.

The spiral equation connecting the prey's position and the whale's position to mimic the humpback whales' helix-shaped movement can be stated as follows:

$$X(t+1) = D'.e^{bl}.\cos(2\pi l) + X^*(t) \quad \text{Eq. 6}$$

where b is the constant factor responsible for the shape of spirals, l randomly belongs to the interval [−1, 1] and D' shows the distance of the i-th whale to the prey (the best solution achieved until this moment) as stated in Eq 7.

$$D' = |X^*(t) - X(t)| \quad \text{Eq. 7}$$

There is a 50% chance of selecting either the spiral model or the shrinking encircling mechanism to model the two concurrent approaching behaviors during the bubble-net attacking strategy, as stated in Eq. 8.

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D & \text{if } p < 0.5 \\ D'.e^{bl}.\cos(2\pi l) + X^*(t) & \text{if } p \geq 0.5 \end{cases} \quad \text{Eq. 8}$$

where p represents a random real number between [0, 1].

3) Exploration phase - search for prey: In searching for prey during the exploration stage, a whale should move away from a reference whale. The variation of the A vector can be used with random values less than -1 or greater than 1 to force search agents to move away from a reference whale. The mathematical model for the exploration phase is as follows:

$$D = |C \cdot X_{\text{rand}} - X| \quad \text{Eq. 9}$$

$$X(t+1) = X_{\text{rand}} - A \cdot D \quad \text{Eq. 10}$$

As opposed to the exploitation phase, in the exploration phase, the position of a search agent is updated according to a randomly selected search agent rather than the best search agent discovered thus far. The WOA algorithm can conduct a global search with the help of this mechanism and |A| > 1 emphasis on exploration.

Improved Whale Optimization Algorithm (I-WOA)

Wang et al. proposed an improvement of WOA (I-WOA). The improvement objectives are to increase exploration and exploitation potential and decrease the likelihood of entering the local optimum by introducing a nonlinearly modified convergence factor, incorporating a new inertia weight factor, and modifying the execution time of the present optimal individual (Wang, Deng, Zhu, & Hu, 2019).

The performance of basic WOA is improved in three aspects: nonlinear convergence factor, inertia weight factor, and random variation of best search agent. According to Wang et al. the linearly changed convergence factor a in Eq. 5 cannot reflect the real optimizing process of the algorithm and limits the exploration and exploitation ability. They proposed a nonlinearly changed convergence factor as follows:

$$a(t) = \frac{2(1-t/t_{\max})^2}{(1-\mu t/t_{\max})^3} \qquad \text{Eq.11}$$

where μ is the adjustment coefficient and the value is in the interval [15,35].

Inspired by the PSO algorithm, a new inertia weight factor is introduced to enhance exploration and exploitation ability and accelerate convergence speed. The position updated method with inertia weight factor is depicted in the following equation.

$$X(t+1) = \begin{cases} \omega X^*(t) - A.D & \text{if} \quad p < 0.5, |A| < 1 \\ \omega X_{\text{rand}}(t) - A.D & \text{if} \quad p < 0.5, |A| \geq 1 \\ D'.e^{bl}.\cos(2\pi l) + \omega X^*(t) & \text{if} \quad p \geq 0.5 \end{cases} \qquad \text{Eq. 12}$$

where ω is inertia weight factor and is calculated as

$$\omega = \alpha \times rand() \qquad \text{Eq. 13}$$

where α is a number in the interval [0.5, 2.5].

In WOA, during the exploitation process, all search agents move toward the current best search agent. This situation could trap the algorithm in a local optimum if the present optimal solution is local. Wang et al. suggested a stochastic variation of the most effective search agent to lower the likelihood of a local optimum (Wang, Deng, Zhu, & Hu, 2019). Assuming that $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is the current best search agent, one element $x_k (k = 1, 2, \ldots, d)$ from X_i is chosen and is replaced with a random integer in $[l_i, u_i]$ where l_i and u_i is lower and upper bound of x_i, respectively.

WOA for MOFSP

WOA Flowchart

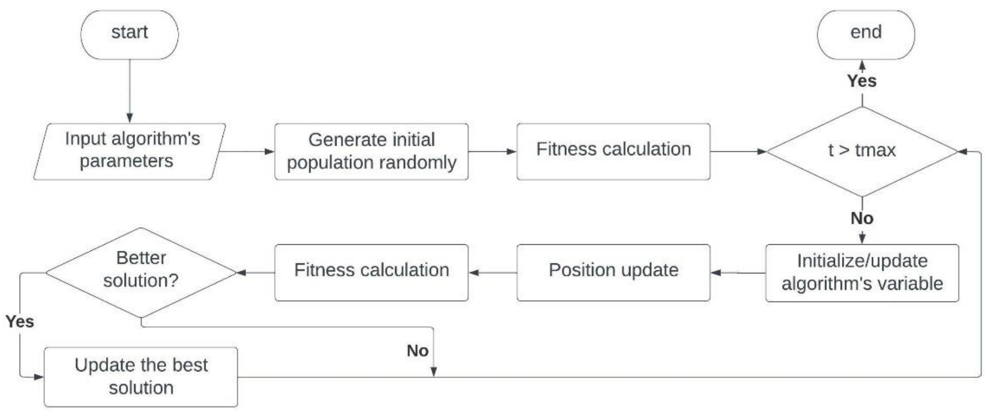

Figure 2. WOA flowchart

In general, WOA and I-WOA have the same framework. Figure 2 illustrates the workflow of the WOA and I-WOA algorithms. After determining the input parameters, the initial population is generated, namely a collection of whales representing candidate solutions to the problem. The whales with the best fitness are stored. Next, the fitness calculation of each whale is carried out. These whales will be subjected to several processes in several iterations. The algorithm variable is recalculated if the maximum iteration has not been reached. Changes in the value of these variables are used as a reference for changing the whales' positions. At this stage, each algorithm may use different formulas as explained in section 2. The fitness calculation is done again using the new whale position. The best candidate is replaced if a better solution is found. After reaching the maximum iteration, the algorithm stops and returns the best solution.

Whale Modeling

Every whale represents a candidate solution of MOFSP, which is a sequence of jobs whose length is the number of jobs. For example, W1 = ⟨1, 3, 2⟩ and W2 = ⟨2, 1, 3⟩ are two whales for a MOFSP with 3 jobs and can be depicted in Figure 3. The value of each whale's *i*-th element states the whale's position in the *i*-th dimension as well.

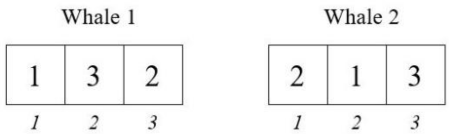

Figure 3. Two whales

Fitness

In this paper, the fitness value used is the objective of MOFSP, which is a combination of makespan and total flow time. For each whale, the formula for its fitness is defined as follows:

$$Fitness = w_m \,.\, C_{max} + w_{tft} \,.\, TFT \qquad \text{Eq. 14}$$

where $0 \leq w_m \leq 1$ and $0 \leq w_{tft} \leq 1$ are real values such that $w_m + w_{tft} = 1$. C_{max} and TFT represent the makespan and the total flow time of the corresponding whale, respectively. The smaller the fitness, the better the quality of a whale.

Distance between Two Whales

Since each whale represents its position, the distance between two whales, w_1 and w_2, can be calculated using the Euclidean Distance:

$$D(w_1,w_2)=\sqrt{\sum_{i=1}^{n}(w_1 - w_2)^2} \qquad \text{Eq. 15}$$

Position Updating

The position of each whale changes with each iteration based on the parameter values. This position update is carried out in the following manner:

1) Store the whale's initial location.
2) Update the vector's elements using Eq. 8, Eq. 10, or Eq. 12.
3) Sort the elements of the vector in ascending order.
4) Set the new location by referencing the element's index that shifted during sorting.

Figure 3 gives an example of position updating.

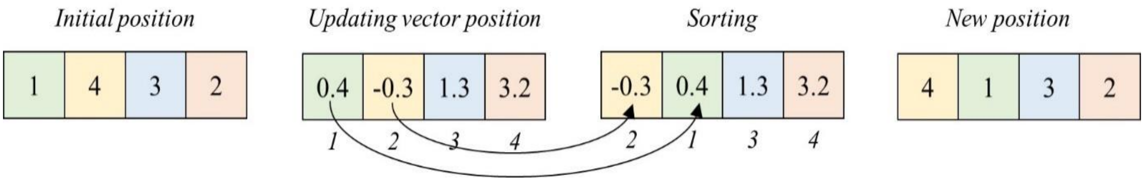

Figure 3. Position update

Computational Experiments

Experiment Setting

As stated previously, this research aims to compare the performance of the WOA and I-WOA algorithms in solving the MOFSP problem. These objectives are divided into two subgoals:

1) determines the effect of algorithm parameters on the performance of WOA and I-WOA.
2) compares the performance of the WOA and I-WOA algorithms.

The parameter to be measured is the combination of objective weights used to determine fitness, as shown in Eq. 14. There are three weight combinations, namely w_1, w_2, and w_3, are used. Table 1 describes the ratio of w_m: w_{tft} for each weight combination.

Table 1. Weight ratio setting

Ratio	w_m	w_{tft}
w_1	0.25	0.75
w_2	0.5	0.5
w_3	0.75	0.25

The other two algorithm parameters are N for the number of whales and t_{max} for the number of iterations. Both are set to 100. Taillard's benchmark was employed in this experiment. There are 12 groups of problem instances. The variation of each group is determined by the number of jobs and machines, as shown in Table 2. Each group contains ten instances. As a result, there are 120 problem instances in total.

Table 2. Problem size

Group	Number of jobs	Number of machines	Group	Number of jobs	Number of machines
c_1	20	5	c_7	100	5
c_2	20	10	c_8	100	10
c_3	20	20	c_9	100	20
c_4	50	5	c_{10}	200	10
c_5	50	10	c_{11}	200	20
c_6	50	20	c_{12}	500	20

For each problem instance, for each parameter combination, we perform 100 computations. So, for each problem and each parameter combination, 100 best solutions are obtained, namely the whales with the best fitness values. In addition, the makespan and total flow time of each of the best solutions are also stored.

From these values, the makespan, total time flow, and fitness values will be generated, which represent the solution of the problem instance. For this, we use two methods. The first way is to calculate the average of 100 fitness, makespan, and total flow time values. Whereas for the second approach, we take the median values instead of the average values. This approach is considered due to the randomness factor applied by each algorithm.

Results and Analysis

To determine the effect of weight ratio, for each solution of a problem, we determine whether the ratio produces the best makespan value, total flow time value, or a combination of the best makespan and total flow time values at once.

Table 3 shows the computation results generated by the WOA algorithm with the average value approach. The columns are grouped into four types. The first column informs the instances group. The second, third, and fourth groups inform the values related to makespan (MS), total flow time (TFT), and the combination of makespan and total flow time (MS-TFT). Each second, third, and fourth group consists of three columns.

Every column represents the weight ratio w_1, $w_{2,}$ and w_3 as previously explained. Each table entry (except the one from the first column) represents how often the corresponding weight ratio yields the best results in its corresponding instance group. For example, the first-row states that for the c1 instance group, w_3 and w_1 are the best weight ratio relative to makespan and total time flow, respectively. However, no weight ratio simultaneously yields the best makespan and total time flow for this case.

The last row summarizes the results. The best ratio relative to makespan alone is w_3 and the best ratio relative to total flow time alone is w_1, whereas for both objectives, the w_3 is the best ratio. Figure 5 illustrates this summary. The same explanation applies to the other three tables: Table 4, Table 4, and Table 6. The computation results provided by the modified WOA algorithm with the average value technique are shown in Table 4 and the summary is illustrated in Figure 6. The best ratio relative to makespan alone is w_3 and the best ratio relative to total flow time alone is w_1, whereas for both objectives, the w_3 is the best ratio.

Table 3. WOA computation results using the average value approach

Group	MS			TFT			MS-TFT		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	0	0	10	8	2	0	0	0	0
c_2	0	0	10	6	4	0	0	0	0
c_3	0	0	10	6	3	1	0	0	1
c_4	0	3	7	3	3	4	0	1	2
c_5	0	2	8	2	2	6	0	1	6
c_6	1	2	7	2	4	4	0	2	3
c_7	0	5	5	4	3	3	0	2	2
c_8	1	1	8	3	4	3	0	1	2
c_9	3	2	5	7	3	0	3	1	0
c_{10}	0	3	7	4	2	4	0	1	3
c_{11}	1	2	7	4	3	3	1	1	3
c_{12}	4	4	2	2	7	1	0	3	0
Total	10	24	86	51	40	29	4	13	22

Table 4. I-WOA computation results using the average value approach

Group	MS			TFT			MS-TFT		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	0	0	10	6	3	1	0	0	1
c_2	0	0	10	5	4	1	0	0	1
c_3	0	0	10	6	4	0	0	0	0
c_4	0	1	9	5	3	2	0	0	2
c_5	0	1	9	4	5	1	0	0	1
c_6	0	2	8	4	3	3	0	1	2
c_7	2	1	7	3	2	5	0	0	2
c_8	1	1	8	2	2	6	1	0	4
c_9	1	2	7	3	2	5	2	1	4
c_{10}	4	3	3	2	4	4	1	2	2
c_{11}	5	2	3	2	3	5	2	1	2
c_{12}	3	3	4	4	1	5	6	1	3
Total	16	16	88	46	36	38	6	6	24

Table 5. WOA computation results using the median value approach

Group	MS			TFT			MS-TFT		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	0	1	10	5	3	2	0	1	2
c_2	0	0	10	6	3	1	0	0	1
c_3	0	0	10	7	1	2	0	0	2
c_4	1	3	6	3	5	2	1	1	0
c_5	3	2	5	4	3	3	0	0	1
c_6	0	3	7	2	5	3	0	2	3
c_7	0	4	6	5	1	4	0	1	3
c_8	0	2	8	2	2	6	0	1	5
c_9	2	1	7	5	3	2	1	1	2
c_{10}	1	3	6	5	2	3	1	1	2
c_{11}	1	1	8	2	5	3	1	1	3
c_{12}	4	6	0	3	6	1	2	4	0
Total	12	26	83	49	39	32	6	13	24

Table 6. I-WOA computation results using the median value approach

Group	MS			TFT			MS-TFT		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	0	0	10	4	4	2	0	0	2
c_2	0	0	10	3	6	1	0	0	1
c_3	0	0	10	7	3	1	0	0	1
c_4	0	0	10	5	3	2	0	0	2
c_5	0	1	9	4	4	2	0	0	2
c_6	0	3	7	2	6	2	0	2	1
c_7	1	4	5	1	1	8	0	0	3
c_8	2	4	4	1	5	4	0	1	0
c_9	2	4	4	2	3	5	1	2	3
c_{10}	5	2	3	0	1	9	0	0	3
c_{11}	4	5	2	4	2	4	1	2	1
c_{12}	4	1	5	2	1	7	1	0	4
Total	18	24	79	35	39	47	3	7	23

The computation results provided by the basic WOA with the median value technique is given in Table 5. Similarly, to the preceding cases, the best ratio for makespan with the median technique is w_1, for total flow time is w_3, and for both objectives is w_3. Figure 7 shows the summary of Table 5.

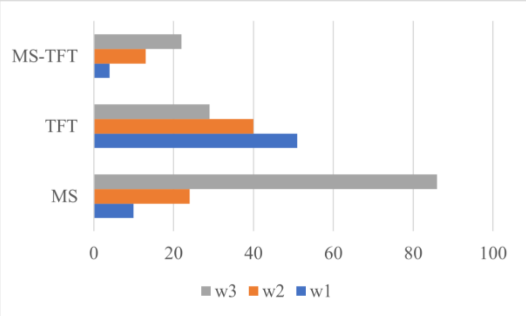

Figure 5. WOA weight ratio effect using average value approach.

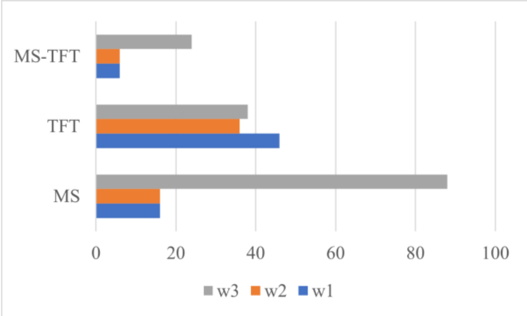

Figure 6. I-WOA weight ratio effect using average value approach.

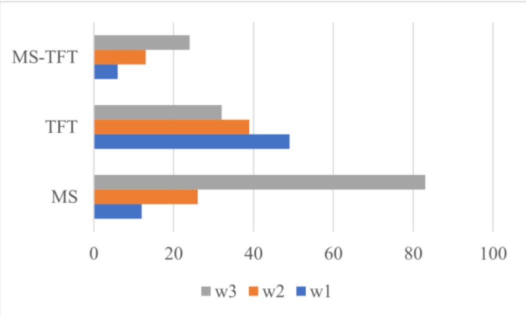

Figure 7. WOA weight ratio effect using median value approach.

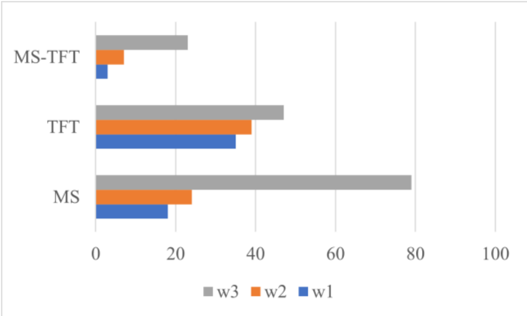

Figure 8. I-WOA weight ratio effect using median value approach.

Table 6 shows the computation results generated by the improved WOA using the median value technique. In contrast to the preceding three cases, the best ratio for time flow ratio is w_3. This condition is shown by the last column in Table 6 and furthermore illustrated in Figure 8. The four tables above show that the effect of the weight ratio varies depending on the group. Table 3, Table 4, and Table 5 all follow a similar pattern, while Table 6 is different, namely the effect on the total flow time.

Furthermore, to compare the performance of the basic and improved WOA algorithms, the results of the computations generated by WOA with the average value approach and the median value approach are juxtaposed with the corresponding results of improved WOA. The comparisons are given in Table 7 and Table 8. Each table entry represents the number of the values generated by the WOA algorithm that are better than the corresponding values generated by the I-WOA algorithm. For all columns, from 120 values, the WOA algorithm produces more than 60 better values (more than half). As a summary, Figure 9 presents the comparison between the performance of WOA and I-WOA in the percentage version. It can be concluded that basic WOA performs better than I-WOA.

Table 7. WOA results with the average value approach

Group	MS			TFT			Fitness		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	5	4	3	5	6	5	5	6	5
c_2	4	4	5	4	4	4	4	4	4
c_3	4	5	2	6	6	5	6	6	5
c_4	8	5	2	6	6	5	6	5	6
c_5	5	7	7	4	4	6	4	5	6
c_6	7	4	5	6	6	4	6	6	4
c_7	6	6	7	6	6	4	6	6	4
c_8	5	7	5	8	8	7	8	8	7
c_9	4	7	6	3	4	5	3	4	5
c_{10}	8	7	9	9	9	7	9	9	7
c_{11}	6	3	6	6	6	6	6	5	7
c_{12}	8	9	5	10	9	5	10	9	5
Total	70	68	67	73	73	64	73	73	65

Table 8. I-WOA results with the average value approach

Group	MS			TFT			MS-TFT		
	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
c_1	5	4	3	7	6	5	5	6	5
c_2	4	1	6	3	4	3	5	4	4
c_3	5	4	2	6	7	5	6	7	5
c_4	8	6	4	6	5	6	6	5	6
c_5	6	6	5	4	4	5	4	4	6
c_6	6	4	6	5	7	4	5	6	5
c_7	7	7	7	7	6	4	6	6	4
c_8	6	6	6	8	8	7	8	8	7
c_9	4	4	6	4	5	5	2	5	5
c_{10}	8	7	10	8	8	7	8	8	7
c_{11}	5	5	6	6	6	6	6	5	7
c_{12}	8	7	5	10	9	7	10	9	6
Total	72	61	66	74	75	64	71	73	67

Similarly, Table 8 shows the full comparison results for each group using the median value approach, and Figure 10 shows the summary. The same conclusion is obtained for the approach with the median value. Once again, WOA performs better than I-WOA in general.

Conclusion

In this paper, we have discussed how to solve the Multi-objective Flow Shop Scheduling Problem using the basic WOA algorithm and improved WOA with the objectives of minimizing makespan and total flow time. The performance of the two algorithms is compared by conducting several experiments. In addition, the effect of the objective weight ratio on the performance of each algorithm was also investigated. The conclusions obtained by this study are that the objective weight ratio affects the performance of WOA and I-WOA and the best objective ratio is MS:TFT = 75:25. The second conclusion is that WOA provides better performance than I-WOA.

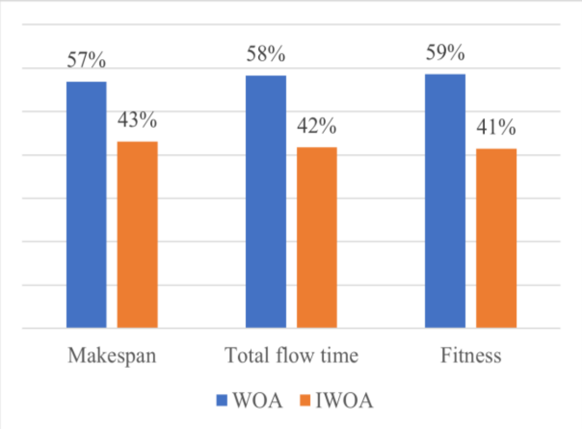

Figure 9. Performance comparison with the average value approach.

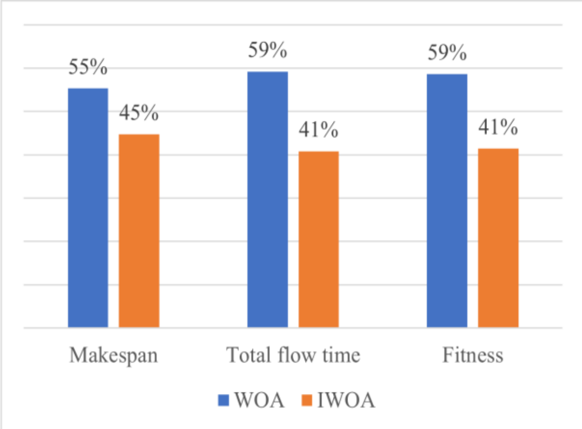

Figure 10. Performance comparison with the median value approach.

Only one variation of the basic WOA was investigated in this study. Many other variants of WOA have been proposed by researchers. There are numerous options for comparing the performance of various algorithms. The performance of WOA in comparison to other metaheuristics, such as the Bee Colony Algorithm lain (Halim & Nugraheni, 2021), is interesting. Hyper-heuristic techniques can also be used in scheduling solutions, such as (Nugraheni & Abednego, 2016b; Nugraheni & Abednego, 2017). The application of WOA with hyper-heuristics is also worth investigating. In addition, research into the applicability of WOA to other optimization issues is also challenging.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Acknowledgements

* This article was presented as an oral presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 16-19, 2022.

* This work is supported by Institution of Research and Community Services, Parahyangan Catholic University with Contract Nr. III/LPPM/2022-02/78-P.

References

Alimoradi, M. (2021). *A report on the whale optimization algorithm (WOA).* https://doi.org/10.13140/RG.2.2.30154.90563.

Alnowibet, K. A., Shekhawat, S., Saxena, A., Sallam, K. M., & Mohamed, A. W. (2022). Development and applications of augmented whale optimization algorithm. *Mathematics*, *10*(12), 2076.

Cao, D., Xu, Y., Yang, Z., Dong, H., & Li, X. (2022). An enhanced whale optimization algorithm with improved dynamic opposite learning and adaptive inertia weight strategy. *Complex & Intelligent Systems*. https://doi.org/10.1007/s40747-022-00827-1.

Chakraborty, S., Saha, A.K. Sharma, S., Mirjalili, S., & Chakraborty, R. (2020). A novel enhanced whale optimization algorithm for global optimization. *Computers & Industrial Engineering. 153*, 107086. https://doi.org/107086.10.1016/j.cie.2020.107086.

Demir, A., & Gelen, M. (2021). A new approach to solving multi-objective flowshop scheduling problems: A multi-MOORA based genetic algorithm. *Journal of Engineering Research. 9*(4A). 191-200. https://doi.org/10.36909/jer.8307.

Halim, Y., & Nugraheni, C. E. (2021). A bee colony algorithm based solver for flow shop scheduling problem. *JOIV: International Journal on Informatics Visualization*, *5*(2), 170-176. https://doi.org/10.30630/joiv.5.2.491.

Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H., & Too, J. (2021). Boosted whale optimization algorithm with natural selection operators for software fault prediction. *IEEE Access*, *9*, 14239-14258.

Liang, X., Xu, S., Siwen, Liu, Y., & Sun, L. (2022). *A modified whale optimization algorithm and its application in seismic inversion problem. Mobile Information Systems. 2022*. https://doi.org/1-18. 10.1155/2022/9159130.

Lin, C.-C., Wu, Z.-X., Huang, K.-W., & Li, Y.-M. (2018). *A hybrid whale optimization algorithm for flow shop scheduling problems*. *Proceeding of the Fourth International Conference on Electronics and Software Science (ICESS2018*), Japan.

Lu, C., Xiao, S., Li, X., & Gao, L. (2016). An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. *Advances in Engineering Software*, *99*, 161-176. https://doi.org/10.1016/j.advengsoft.2016.06.004.

Mirjalili, S. & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *Volume 95*, 51-67, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2016.01.008.

Nadimi-Shahraki, M. H., Zamani, H., & Mirjalili, S. (2022). Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study. *Computers in Biology and Medicine*, *148*, 105858. https://doi.org/10.1016/j.compbiomed.2022.105858.

Ning, G. Y., & Cao, D. Q. (2021). Improved whale optimization algorithm for solving constrained optimization problems. *Discrete Dynamics in Nature and Society,* 1-13. https://doi.org/10.1155/2021/8832251.

Nugraheni, C. E., & Abednego, L. (2016a). A comparison of heuristics for scheduling problems in textile industry. *Jurnal Teknologi*, *78*, 99-104. https://doi.org/10.11113/jt.v78.9034

Nugraheni, C. E., & Abednego, L. (2016b). On the development of hyperheuristics-based framework for scheduling problems in textile industry. *International Journal of Modeling and Optimization*, *6*, 272-276. https://doi.org/10.7763/IJMO.2016.V6.539.

Nugraheni, C. E., & Abednego, L. (2017). A tabu-search based constructive hyper-heuristics for scheduling problems in textile industry. *Journal of Industrial and Intelligent Information*, *5*(2). https://doi.org/ 10.18178/jiii.5.2.23-27.

Rana, N., Latiff, M. S. A., Abdulhamid, S. I. M., & Chiroma, H. (2020). Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments. *Neural Computing and Applications*, *32*(20), 16245-16277. https://doi.org/10.1007/s00521-020-04849-z.

Robert, R.B. &, Rajkumar, R. (2019). A hybrid algorithm for multi-objective optimization of minimizing makespan and total flow time in permutation flow shop scheduling problems. *Information Technology and Control. 48*, 47-57. https://doi.org/10.5755/j01.itc.48.1.20909.

Schulz, S., Buscher, U., & Shen, L. (2020). Multi-objective hybrid flow shop scheduling with variable discrete production speed levels and time-of-use energy prices. *Journal of Business Economics*, *90*(9), 1315-1343. https://doi.org/10.1007/s11573-020-00971-5.

Singh, H., Oberoi, J. S., & Singh, D. (2021). Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review. *RAIRO-Operations Research*, *55*(1), 27-50. https://doi.org/10.1051/ro/2020055.

Wang, Z., Deng, H., Zhu, X., & Hu, L. (2019). Application of improve whale optimization algorithm in muti-resource allocation. *Int. J. Innovative Comput*, *15*(3), 1049–1066.

Author Information

Cecilia Esti Nugraheni
Informatics Dept.
Parahyangan Catholic University
Jl. Ciumbuleuit 94, Bandung, 40141
Indonesia
Contact e-mail: *cheni@unpar.ac.id*

Craven Sachio Saputra
Informatics Dept.
Parahyangan Catholic University
Jl. Ciumbuleuit 94, Bandung, 40141
Indonesia

Luciana Abednego
Informatics Dept.
Parahyangan Catholic University
Jl. Ciumbuleuit 94, Bandung, 40141
Indonesia

To cite this article:

Nugraheni, C. E., Saputra, C. S., & Abednego, L. (2022). Whale optimization algorithms for multi-objective flowshop scheduling problems. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 21,* 441-451.