Received: November 17, 2023. Revised: February 12, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Modified Mountain Gazelle Optimizer for Minimizing Energy Consumption on No-Wait Permutation Flow Shop Scheduling Problem

Dana Marsetiya Utama[1] **Bagus Dzulhi Sanjaya[1]*** **Adhi Nugraha[1]**

[1]*Department of Industrial Engineering, University of Muhammadiyah Malang, Malang, Indonesia*
* Corresponding author's Email: balhisanjaya@gmail.com

Abstract: In the context of growing global energy demand, the industrial sector has become one of the significant contributors to the world's energy consumption. To face this challenge, scheduling has been identified as one of the potential methods to reduce energy consumption in industrial operations. This article introduces the mountain gazelle optimizer (MGO) algorithm as a solution to solve the no-wait flow shop scheduling problem with a focus on the main objective of minimizing energy consumption. The research involves comparing the performance of the MGO algorithm with popular algorithms such as grey wolf optimizer (GWO), particle swarm optimization (PSO), and genetic algorithm (GA). In addition, this research also compares the proposed MGO algorithm with the latest advanced algorithms, such as coati optimization algorithm (COA) and fire hawk optimizer (FHO). In this work, the six algorithms were tested on three different scheduling cases by repeating the process 30 times, using a population of 200 and 200 iterations to minimise energy consumption. The performance comparison between these algorithms was analyzed using the One-Way ANOVA statistical test. Based on the results, MGO outperforms GWO, PSO, GA, COA, and FHO algorithms in solving scheduling problems, with the primary function of minimizing energy consumption in Cases 1, 2, and 3. In addition, based on the convergence curve, the MGO algorithm has a better convergence curve compared to GWO, PSO, GA, COA, and FHO; it shows that the intestinal algorithm can reach the optimal solution faster and more stable during iterations than the GWO, PSO, GA, COA, and FHO algorithms. This finding confirms that the MGO algorithm has the potential to be an effective alternative in the scheduling optimization process, significantly reducing energy consumption in the industrial sector.

Keywords: Mountain gazelle optimizer, Flow shop, Energy consumption, No-wait, Scheduling.

1. Introduction

In recent years, worldwide, the energy demand has increased significantly. One of the most critical resources for the industrial sector is energy [1-3]. Excessive energy use accounts for half of the world's total energy consumption [4]. Excessive energy use leads to environmental damage, such as global warming and greenhouse gases [5, 6]. Therefore, the industrial sector must take action to reduce energy use with no waiting time [7]. Generally, energy consumption and no waiting time occur during concrete production [8]. One scheduling method that can be applied to overcome this is the no-wait flow shop scheduling problem, where this method requires that each job can only be processed by one machine at a time [9]. Meanwhile, essential metaheuristic algorithms are used in the no-wait flow shop scheduling problem [10]. This problem refers to the continuous flow of jobs through different machines, where jobs, once started, must have continuous processes [11]. Metaheuristic algorithms are widely used to tackle hard-to-solve optimization problems [12-14]. Several new advanced metaheuristic procedures have also been proposed, such as the Coati Optimization Algorithm (COA) [15], swarm magnetic optimizer [16], walk-spread algorithm walk-spread algorithm [17], and four directed search algorithm [18]. Metaheuristic algorithms are handy for solving scheduling problems, such as the no-wait flow shop scheduling problem, where they can find near-optimal solutions by applying reasonable computational time [19]. The mountain gazelle optimizer is a metaheuristic algorithm used to solve this problem. Mountain Gazelle optimizer has been proven effective in solving various optimization problems, such as mathematical function optimization, system parameter optimization, machine design optimization, and frame structure optimization [20].

Previous studies of problems on no-wait permutation flow shop scheduling have suggested various approaches to achieve different objectives. K. Gao [21] proposed a heuristic for the no-wait flow shop scheduling problem to minimize the total flow time. In their research, M. Seido Nagano [22] proposed constructive heuristics on the no-wait flow shop scheduling problem with sequence-dependent setup times to minimize total flow time. H. Ye [23] proposed heuristics for the no-wait flow shop scheduling problem to minimize the total completion time. Laha dan Chakraborty [24] proposed a constructive heuristic on the no-wait flow shop scheduling problem to minimize the makespan. Y. Wang [25] proposed an iterated greedy heuristic on the no-wait permutation flow shop scheduling problem to minimize the makespan. Meanwhile, Lin dan Ying [26], in their research, used a MIP (Mixed Integer Programming) based search procedure to minimize the makespan in the no-wait flow shop scheduling problem. Tavakkoli-Moghaddam, Rahimi-Vahed, and Mirzaei [27] used an immune algorithm to reduce completion time and tardiness in the no-wait flow shop scheduling problem. M. Rojas-Santiago [28] proposed an ant colony optimization approach to the no wait flow shop scheduling problem to minimize makespan. Several studies have also concentrated on minimising energy consumption for no-wait permutation flow shop scheduling problems. Q.-q. Zeng [29] proposed a non-dominated sorting genetic algorithm to minimize the makespan and total energy consumption. F. Zhao [30] proposed a reinforcement learning-driven brainstorm optimization algorithm to minimize completion time and total energy consumption. F. Zhao [31] proposed a two-stage cooperative evolutionary algorithm to minimize makespan and total energy consumption in the no-wait flow shop scheduling problem. F. Zhao [32] used an improved iterative greedy algorithm to minimize completion time and energy consumption. Zhao, Jiang and Wang [33] used a cooperative meta-heuristic algorithm based on Q-Learning to solve the total energy consumption.

The existing literature on no-wait flow shop scheduling reveals a noticeable research gap in energy consumption. While previous studies have predominantly concentrated on minimizing completion time and total flow time as primary objectives, exploring energy efficiency optimization in this context still needs to be explored. The existing research lacks in-depth investigations into how energy consumption can be effectively minimized within the no-wait flow shop scheduling framework. This paper proposes a new advanced MGO procedure inspired by the Gazelle's hierarchical and social life in the wild. The MGO algorithm uses gazelle movement in searching for food and avoiding predators to find the optimal solution to optimization problems. Furthermore, a significant void exists in the literature regarding the application and efficacy of the mountain gazelle optimizer (MGO) algorithm in addressing scheduling problems, particularly in the no-wait permutation flow shop scheduling problem.

This dearth of research limits our understanding of the energy dynamics in no-wait flow shop scheduling. It overlooks the potential contributions of advanced optimization algorithms like MGO in enhancing scheduling solutions. Integrating MGO in this research is a new opportunity to bring innovation and improve the solution's performance while considering a crucial factor in energy consumption. Therefore, this research will pave the way for a more comprehensive understanding of optimising energy efficiency in no-wait flow shop scheduling. This research aims to propose the mountain gazelle optimizer (MGO) algorithm to solve the no-wait permutation flow shop scheduling problem with the objective function of minimizing the total energy consumption. MGO is a new algorithm proposed by B. Abdollahzadeh [34]. In the context of classical MGO, the MGO algorithm is used to solve problems where the decision variables are real numbers. However, the no-wait scheduling problem in flow shops, considered a discrete problem, requires a customized approach. Therefore, the MGO algorithm undergoes an innovative modification in this study by integrating the large rank value (LRV) principle. This modification aims to overcome the incompatibility of the MGO algorithm with discrete problems, particularly in the context of no-wait permutation flow shop scheduling. By embedding the LRV principle, the MGO algorithm can effectively convert real numbers into discrete permutation number representations, demonstrating the originality of the proposed technique to handle this particular problem. Thus, this approach is expected to improve MGO's performance in solving discrete scheduling problems.

This research also tried to compare the proposed procedure of the MGO algorithm with popular algorithms such as the grey wolf optimizer algorithm (GWO) [35], particle swarm optimization (PSO) [36], and genetic algorithm (GA) [37, 38]. This research proposes the MGO algorithm to handle the no-wait permutation flow shop scheduling problem but also conducts a comparison with the latest state-of-the-art algorithms. Furthermore, it compares the performance of the MGO algorithm with two innovative algorithms, namely the coati optimization algorithm (COA) [39] and the fire hawk optimizer (FHO) [40]. By conducting this comparison, this research seeks to provide an in-depth understanding of the relative advantages and disadvantages of the MGO algorithm compared to recent algorithms in solving scheduling problems. Through the MGO approach, which has proven to be effective in various optimization contexts, this research can significantly contribute to improving production system performance and saving energy resources. By integrating the advantages of the MGO algorithm with the primary objective of this research, it is expected to provide reliable and efficient solutions to complex challenges in the production schedule planning domain.

This research makes significant contributions through several aspects. First, this work proposes the MGO algorithm as a solution to handle the no-wait permutation flow shop scheduling problem. This algorithm is designed to improve scheduling efficiency between jobs without waiting. Secondly, this research enriches the scholarship on the problem by incorporating a new objective function, minimizing energy consumption. It broadens the scope of our understanding of the no-wait permutation flow shop scheduling problem. It provides new insights into the scheduling process's sustainability and energy efficiency aspects. Finally, this research achieves an essential contribution by analyzing and comparing the performance of the modified MGO algorithm with several other algorithms, such as GWO, PSO, GA, COA, and FHO. The results of this comparison provide valuable insights regarding the effectiveness and relative advantages of the MGO algorithm in the context of scheduling problem-solving. Thus, this research not only offers an innovative solution to the problem but also contributes to our understanding of the application and performance improvement of algorithms in the context of scheduling. MGO is expected to provide an optimal or near-optimal solution in optimizing energy usage in a flow shop environment. By reducing the total energy required in the scheduling process, this research can positively impact the sustainability of company operations, reduce environmental impact, and improve resource use efficiency. The results of this research are expected to be the basis for further development in the context of environmentally friendly production scheduling.

The rest of this article is organized in the following structure: section 2, entitled "The Proposed Method," will detail the procedure of the MGO proposed in this study. In section 3, which focuses on "The Method," we will describe the data used and the execution of experiments to evaluate the performance of the MGO. Section 4, entitled "Results," will present and analyze the experimental results, including comparing MGO's performance with other algorithms. Finally, the paper will conclude in the Conclusion Section.

2. The proposed method

This section describes the assumptions, notations, and mathematical models used in the research. The problem, in this case, has the following assumptions: (i) a job that has been started on a machine cannot be stopped before completion (no pre-emption), (ii) the job consists of a consistent sequence of operations and the machine is in ready condition, (iii) all jobs and machines are ready at time t=0, (iv) each machine performs only one job at a time, (v) the next job can be performed if the machine has completed the previous job, (vi) preparation time is included in the processing time.

In this study, the notation and mathematical formulation of NWPFSP to minimize total energy consumption refers to Yüksel's research [41]. The notation used is as follows:

N	:	set job
M	:	machine set
L	:	Set speed level
k	:	job index ($1 \leq k \leq \|N\|$)
r	:	machine index ($1 \leq r \leq \|M\|$
l	:	speed level index ($1 \leq l \leq \|L\|$
P_{ur}	:	processing time of job - u on the machine - r
Y_l	:	the conversion factor for process speed level
φ_r	:	the conversion factor for idle time on the machine
τ_r	:	machine energy (kW)
θ_r	:	idle time on machine - r
π	:	job permutation
C_{ur}	:	Completion time on machine - r
C_{max}	:	makespan value
Q	:	permutations to avoid overlap
TEC	:	total energy consumption
TSM	:	territorial solitary males
$male_{gazelle}$	:	position vector of the best global solution
BH	:	vector of the impact factor of young males

ri_1 and ri_2	:	parameter random integers 1 or 2
F	:	Dimensions of the issue
$X(t)$	:	position of the gazelle vector in the current iteration
Cof_r	:	randomly selected coefficient vector update in each iteration
X_{ra}	:	random solution young male in the interval of ra
M_{pr}	:	average number of search agents
r_1 and r_2	:	Random values between 0 and 1
N_1	:	random number from the standard distribution
exp	:	function exponential
$MaxIter$	:	total iterations
$Iter$	:	current number of iterations
r_3, r_4, and $rand$	:	random number in the range of 0 and 1
N_2, N_3 and N_4	:	random number in the normal range and the dimensions of the problem
cos	:	function cosine
MH	:	maternity herds
$Cof_{2,r}$ and $Cof_{3,r}$	:	randomly selected co-efficient vectors
ri_3 and ri_4	:	integer and random numbers 1 or 2
X_{rand}	:	Vector position of a gazelle that is randomly selected from the entire population
BMH	:	bachelor male herds
ri_5 dan ri_6	:	Integers 1 or 2 that are chosen randomly
MSF	:	migration to search for food
ub dan lb	:	upper and lower limits
r_7	:	An integer between 0 and 1 chosen at random

Furthermore, there is a mathematical formulation of mixed integer linear programming (MILP) for solving the case of energy consumption minimization:

Objectives

$$\text{Minimize TEC} \quad (1)$$

Constraint

$$C_{ul} \geq \sum l \in L \times P_{ur} \times \gamma_{i1l} \quad (2)$$

$$C_{ur} - C_{u,r-1} \geq \sum l \in L \times P_{ur} \times \gamma_{u1l} \quad (3)$$

$$C_{ur} - C_{k,r-1} + Q \times D_{uk} \geq \sum l \in L \times P_{ur} \times \gamma_{u1l} \quad (4)$$

$$C_{ur} - C_{k,r-1} + Q \times D_{uk} \leq Q - \sum l \in L \times P_{kr} \times \gamma_{k1l} \quad (5)$$

$$C_{max} \leq C_{uM} \quad (6)$$

$$C_{ur} - C_{u,r-1} \leq \sum l \in L \times P_{ur} \times \gamma_{u1l} \quad (7)$$

$$l \in L^{yurl} = 1 \quad (8)$$

$$y_{url} = y_{u,r+1,l} \quad (9)$$

$$\theta_r = C_{max} - \sum u \in N \sum l \in L \times P_{ur} \times y_{url} \quad (10)$$

$$TEC = \sum u \in N \sum r \in M \sum l \in L \times \frac{P_{ur} \times \tau_r \times y_l}{60} \partial_{url} + \sum r \in M \frac{\theta_r \times \tau_r \times \varphi_r}{60} \quad (11)$$

$$\partial_{Url} \in \{0,1\} \qquad \forall u \in N, \forall r \in M, \forall l \in L \quad (12)$$

$$C_{ur} \geq 0 \qquad \forall u \in N, \forall r \in M \quad (13)$$

$$D_{ik} \in \{0,1\} \qquad \forall u, k \in N: k > u \quad (14)$$

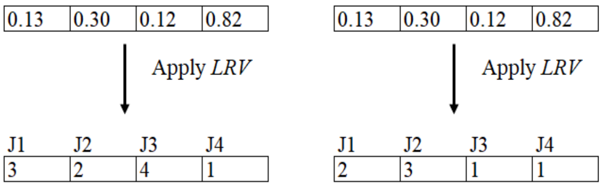
Eq. (1) represents the objective function of minimizing energy consumption. Eq. (2) ensures that the completion time of each job on machine 1 cannot be shorter than its processing time on machine 1. Eq. (3), like Eq. (2), ensures that the completion time of each job on the previous machine r cannot be less than the sum of its processing time on machine r plus its completion time on machine r - 1. In the sequence, pairwise disjunctive Eqs. (4) and (5) ensure that job u follows job k or job k follows job u but not both. Eq. (6) calculates the maximum completion time (makespan) of all jobs on the last machine. Then Eq. (7) ensures that each job on machine r cannot be completed later than the sum of its processing time on machine r plus the makespan on the previous machine r - 1. As a result, a no-wait requirement is provided along with Eqs. (3) and (4). Specifically, the processing time of each job on machine r is determined by the difference between its completion time on machine r and machine r - 1. The constraint saves one assignment of a speed factor for each job Eq. (8). Furthermore, the constraint uses a job-based speed scaling technique, which maintains the same speed level for all jobs on all machines Eq. (9). Eqs. (10) and (11), respectively, calculate machine idle

Received: November 17, 2023. Revised: February 12, 2024.

$$\begin{bmatrix} 0.13 & 0.30 & 0.12 & 0.82 \\ 0.62 & 0.44 & 0.89 & 0.77 \\ 0.56 & 0.73 & 0.73 & 0.95 \end{bmatrix} \qquad \begin{bmatrix} 0.13 & 0.34 & 0.82 & 0.82 \\ 0.62 & 0.83 & 0.44 & 0.44 \\ 0.56 & 0.77 & 0.95 & 0.95 \end{bmatrix}$$

Accepted Population *Rejected Population*

Figure. 1 Illustration of gazelle random population initiation



(a) (b)

Figure. 2 Illustration of LRV application: (a) correct job permutation and (b) incorrect job permutation

time and total energy usage (in kWh). Finally, Eqs. (12), (13), and (14) define binary variables and sign restrictions Eq. (14).

The proposed procedure in this study consists of several essential steps to optimize the no-wait permutation flow shop scheduling problem. The MGO algorithm is presented as the primary method to solve this problem. MGO is a metaheuristic algorithm inspired by a mountain gazelle's social behaviour and life. The basic concepts of mountain gazelle social and group life are used to develop the MGO algorithm by modelling essential aspects such as territoriality and the distance separating their territories. The next stage involves optimization operations that reflect critical elements in mountain gazelle life, including territorial solitary males, maternity herds, bachelor male herds, and migration to search for food. Each iteration of the algorithm aims to produce the best solution. In addition, to modify the position of the MGO in the schedule sequence, this study introduces the Large Rank Value (LRV) procedure as proposed by [42, 43].

Large rank value (LRV)

Gazelle population positions are randomly generated. This research proposes that the population position should ensure that no number is repeated. Figure 1 shows an illustration of the same population initialization. It generates a population that cannot be converted into a permutation sequence. In converting real numbers to permutation sequences, this research proposes to convert gazelle positions into permutation jobs by applying LRV. LRV is seen as a successful way to convert rael values into a sequence of permutation jobs [44]. In LRV, continuous values are sorted from largest to smallest [45]. Fig. 2 describes an illustration of the LRV application.

Stage 1: Territorial solitary males

When male mountain gazelles reach adulthood and become strong enough, they create a solitary territory. They are highly territorial, and great distances separate the territories. The battle between adult male gazelles takes place over the territory or possession of the female. The young males try to occupy the female's territory; on the other hand, the adult males try to protect their environment. Eq. (15) has been used to model the adult male territory.

$$TSM = male_{gazelle} \\ -\left|\left(ri_1 \times BH - ri_2 \times X(t)\right) \times F\right| \times Cof_r \quad (15)$$

In Eq. (15), $male_{gazelle}$ is the position vector of the best global solution (adult male). The parameters ri_1 and ri_2 are random integers 1 or 2. BH is the young male herd coefficient vector, calculated using Eq. (16). F is computed using Eq. (17). Cof_r is also a randomly selected coefficient vector updated in each iteration and used to increase the search capability, calculated using Eq. (18).

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lceil r_2 \rceil,\ ra\left\{\left\lceil \frac{N}{3} \right\rceil \dots N\right\} \quad (16)$$

In Eq. (16), X_{ra} is a random solution (young male) in the interval of ra. M_{pr} is the average number of search agents $\left\lceil \frac{N}{3} \right\rceil$ which were randomly selected. Also, N is the total number of gazelles, while r_1 and r_2 are random values between 0 and 1

$$F = N_1(D) \times \exp\left(2 - Iter \times \left(\frac{2}{MaxIter}\right)\right) \quad (17)$$

In Eq. (17), in the dimension of the problem, N_1 is a random number from the standard distribution. The exponential function is also known as exp, $MaxIter$ is the total number of iterations, and $Iter$ is the current number of iterations.

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos\left((r_4 \times 2) \times N_3(D)\right), \end{cases} \quad (18)$$

In Eq. (18), a is calculated using Eq. (19). Also, r_3, r_4, and $rand$ are random numbers in the range of 0 and 1. N_2, N_3 and N_4 are random numbers in the normal range and the dimensions of the problem. In the problem dimension, r_4 is also a random number in the range of 0 and 1. Finally, cos represents the cosine function.

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter}\right) \quad (19)$$

Finally, in Eq. (19), $MaxIter$ represents the total iterations, and Iter represents the current number of iterations.

Stage 2: Martenity herds

Martenity herds play an essential role in the life cycle of the mountain gazelle, as these types of packs give birth to solid male gazelles. Male gazelles can also play a role in the delivery of gazelles and young males trying to possess females. This behaviour is formulated using Eq. (20).

$$MH = \left(BH + Cof_{1,r}\right) + \left(ri_3 \times male_{gazelle} - ri_4 \times X_{rand}\right) \times Cof_{1,r} \quad (20)$$

In Eq. (20), BH is the vector of young male impact factors used in calculating Eq. (16). $Cof_{2,r}$ and $Cof_{3,r}$ are a vector of randomly selected coefficients calculated independently using Eq. (18). ri_3 and ri_4 are integers and random numbers 1 or 2. $male_{gazelle}$ is the best (adult male) global solution in the current repetition. Lastly, X_{rand} is a vector of gazelle positions randomly selected from the whole population.

Stage 3: Bachelor male herds

As male gazelles mature, they tend to create territories and possessions over female gazelles. At this time, young male gazelles enter into combat with male gazelles for territory and control over female gazelles, which may be accompanied by violence. Eq. (21) is used to formulate gazelle behaviour mathematically.

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH\right) \times Cof_r \quad (21)$$

In Eq. (21). $X(t)$ is the position vector of the Gazelle at the current iteration. D is calculated using Eq. (22). ri_5 and ri_6 are randomly selected integers 1 or 2. $male_{gazelle}$ is the vector position of the male Gazelle (best solution). In addition, BH is the impact factor of the young male herd, which is calculated using Eq. (16). Cof_r is a randomly selected coefficient vector calculated and used using equation (18).

$$D = \left(|X(t)| + \left|male_{gazelle}\right|\right) \times (2 \times r_6 - 1) \quad (22)$$

In Eq. (22), $X(t)$ and $male_{gazelle}$ are the vector position of the Gazelle at the current iteration, and the vector position is the best solution (mature male). r_6 is also a random number between 0 and 1.

Stage 4: Migration to search for food

Mountain gazelles constantly search for food sources and travel long distances to obtain food and migrate. On the other hand, mountain gazelles have high running speed and good jumping power. Eq. (23) has been used to formulate gazelle behaviour mathematically.

$$MSF = (ub - lb) \times r_7 + lb \quad (23)$$

In Eq. (23), ub and lb are the upper and lower bounds of the problem, respectively. Finally r_7 is an integer between 0 and 1 chosen randomly.

The four TSM, MH, BMH, and MSF mechanisms are applied to all gazelles to produce new generations of gazelles. A new era is added to the total population, and each generation equals one replication. Moreover, all gazelles are arranged in ascending order at the end of each era. The best gazelles, which have high quality, promising solutions, and cost less, are preserved in the population. Other gazelles, considered old or weak, are removed from the whole population. The best Gazelle is also considered the adult male Gazelle who owns the territory. Algorithm 1 shows the developed pseudocode of the MGO algorithm.

3. Method

In this study, the research data used three different cases. In Case 1 [46], with a small category with a problem of 10 jobs and 6 machines, process energy's minimum and maximum values are 0.2709 kWh and 0.9409 kWh, respectively. In contrast, idle energy's minimum and maximum values are 0.00296 kWh and 0.1007 kWh. Then, in Case 2 [47], with a medium category with a problem of 30 jobs and 10 machines, the minimum and maximum values of process energy are 0.1825 kWh and 0.9815 kWh. At the same time, the minimum and maximum values of idle energy are 0.0033 kWh and 0.0236 kWh. Finally, in Case 3 [48], with a large category with a problem of 50 jobs and 10 machines, the minimum and maximum values of process energy are 0.1757 kWh and 0.8840 kWh, respectively. In contrast, idle energy's minimum and maximum values are 0.0009 Kwh and 0.0380 Kwh.

In this study, the parameters used in experiments include population and iterations, namely 200 and 200. The experimental procedure on the six

Algorithm 1: Pseudocode of mountain Gazelle optimizer

% Using Mountain Gazelle Optimer (MGO)
Population size N and maximum number of iterations T are inputs
Output: Location and fitness potential of Gazelle's
% Initialization stage
Using $\boldsymbol{X_i} (\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{N})$, create a random population
Use the Large Rank Value (LRV) to convert the MGO position into a job order.
%Determine the fitness level of the Gazelle.
While (the condition to stop is not met) do it
  For (Each Gazelle ($\boldsymbol{X_i}$))
    %Alone male realm
    Using eq. (15), calculate TSM
    %Mother and child herd
    Using Eq. (20), calculate MH
    %Young male herd
    Using Eq. (21), calculate BMH
    %Migration to search for food
    using Eq. (23), calculating *MSF*
    determine the fitness values of *TSM*, *MH*, *BMH*, and *MSF* and then enter them into the habitat.
  End for
  Putting the entire population in ascending order
  update $best_{Gazelle}$
  Keeping the best N Gazelles in maximum population numbers
End while
Return $X_{BestGazelle}$, best fitness

algorithms was run 30 times to find the minimization of energy consumption. The results of each experiment were recorded carefully. The OneWay Anova test was used to find out which method was the most superior and successful among the six algorithms. Experimental results are also shown with the convergence curve. Empirical calculations were performed using R2021a software on Windows 11 with AMD Ryzen 5 4600H and a Radeon Graphics 3.00 GHz processor.

4. Result and discussion

Fig. 3 explains and describes the convergence curve of case 2. The convergence curve is a graph showing an optimisation algorithm's performance increases as iterations progress. This graph shows how quickly the algorithm reaches the optimal solution and stabilizes its performance. This convergence curve compares the performance in solving the no-wait permutation flow shop scheduling problem. Seen in Fig. 3 is the convergence curve in case 2, where the MGO algorithm produces the optimal solution of the GWO, PSO, GA, COA, and FHO algorithms.

Results show that the MGO algorithm exhibits remarkable advantages in exploring the solution space compared to other optimization algorithms, such as GWO, PSO, GA, COA and FHO when applied to the no-wait flow shop scheduling optimization problem focusing on energy consumption minimization. MGO, inspired by the agile behaviour of mountain gazelle, features high explorative ability, which enables this algorithm to navigate complex search spaces efficiently. It means that MGO can find more optimal solutions in a shorter timeframe, reducing the energy consumption required to run the production process. In this regard, MGO emerges as a promising option to improve the efficiency of industrial operations by integrating energy factors into production schedule planning, and further research on the use of MGO in this scope is warranted.

The results of the experiments conducted on three different cases by running each algorithm 30 times have been recorded and are presented in Table 1. In analyzing the results, it can be observed that the MGO algorithm shows significant performance. MGO was able to consume lower average energy compared to GWO, PSO, GA, COA, and FHO algorithms in each case tested. This phenomenon signifies the ability of MGO to provide an optimal solution that considers efficiency in the no-wait flow shop scheduling planning and minimizes the overall energy resource consumption. These results confirm that using the MGO algorithm in this context can substantially positively contribute to operational efficiency and environmental sustainability within the scope of the manufacturing industry.

The statistical analysis results using the one-way ANOVA test in Table 2 show the performance comparison between the MGO algorithm with GWO, PSO, GA, COA and FHO in three cases. Based on the results of multiple comparisons, in case 1, there is a significant difference between the MGO-PSO algorithm. At the same time, MGO-GWO, MGO-GA, MGO-COA and MGO-FHO show similar results. In cases 2 and 3, it was found that the performance of MGO-PSO, MGO-GA, MGO-COA and MGO-FHO algorithms were significantly different, but MGO-GWO showed similar results. These results prove that MGO can be an effective alternative procedure for solving this problem. Furthermore, significant results (Sig) with values > 0.05 indicate similar performance, while values < 0.05 indicate significant differences. It demonstrates that MGO has a good

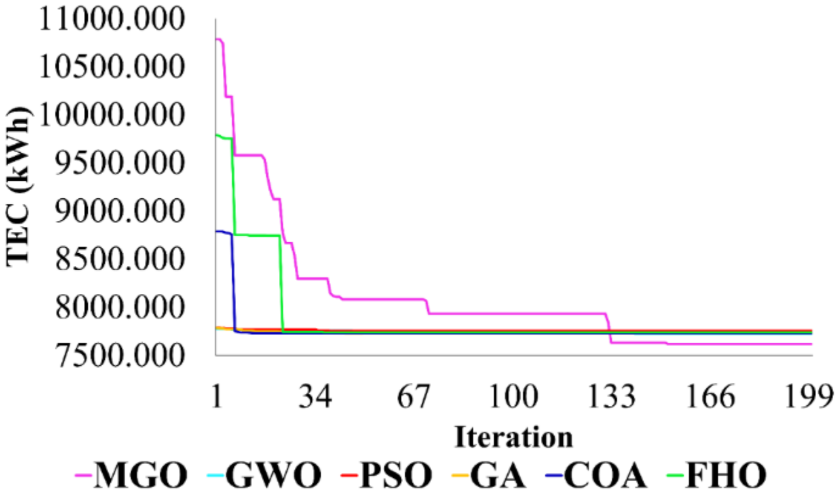


Figure. 3 Convergence curve of case 2

Table 1. The result descriptive statistical of MGO, GWO, PSO, GA, COA, and FHO algorithms

Case	Algorithm	Mean	Std. Deviation	Minimum	Maximum
Case 1	MGO	17305.420	11.539	17301.638	17339.454
	GWO	17317.158	24.498	17301.638	17394.102
	PSO	17340.351	41.596	17301.638	17411.808
	GA	17318.492	19.972	17301.638	17366.122
	COA	17321.314	22.364	17301.638	17401.534
	FHO	17335.904	25.139	17301.638	17426.453
Case 2	MGO	7722.042	5.875	7711.507	7740.151
	GWO	7725.080	7.125	7711.507	7747.356
	PSO	7740.556	9.500	7722.531	7764.195
	GA	7731.364	7.148	7716.021	7749.265
	COA	7736.247	7.136	7717.113	7743.016
	FHO	7738.654	8.785	7720.481	7749.179
Case 3	MGO	12163.833	10.401	12144.867	12184.059
	GWO	12169.163	22.463	12134.075	12225.665
	PSO	12180.481	10.168	12159.990	12201.454
	GA	12176.834	18.552	12143.092	12231.629
	COA	12178.340	20.457	12138.111	12220.974
	FHO	12179.925	24.753	12156.921	12206.992

exploration capability to find stable and consistent optimal solutions across the various test case conditions.

This finding needs to be emphasized that the MGO algorithm has proven superior performance in addressing the issues raised in this study. It is due to the unique characteristics of MGO, which reflect the original behaviour of mountain gazelles. MGO possesses the behaviours of territorial solitary males, maternity herds, bachelor male herds, and migration, which play an essential role in the solution exploration process. These behaviours allow MGOs to incorporate diverse elements in solution search, such as in-depth exploration by combining multiple alternatives and maintaining a diverse population of solutions. Therefore, MGO emerges as one of the most effective and efficient optimization algorithms in solving this problem, delivering high-quality solutions with relatively short computation time. In conclusion, MGO is a powerful tool for exploring and solving the no-wait permutation flow shop scheduling problem based on its characteristics and performance to minimise total energy consumption.

This research has significant theoretical and practical implications. In the academic context, this research successfully introduces the MGO algorithm as an effective solution to the no-wait permutation flow shop scheduling problem, focusing on reducing energy consumption. Experimental results show the superiority of MGO over GWO, PSO, GA, COA and FHO in minimizing energy consumption in three case scenarios, especially on large data. The theoretical implications involve the development of new concepts in the field of scheduling optimization by utilizing the MGO algorithm.

Table 2. The sig level result of multiple comparisons of MGO, GWO, PSO, GA, COA, and FHO algorithms

Algorithm	Case 1	Case 2	Case 3
MGO-GWO	0.3286706	0.4036830	0.5851825
MGO-PSO	0.0000095	0.0000000	0.0007404
MGO-GA	0.2369237	0.0000285	0.0131001
MGO-COA	0.3036936	0.0000369	0.0471975
MGO-FHO	0.2757482	0.0000458	0.0334581

From a practical perspective, this research solves real-world problems related to scheduling no-wait permutation flows by minimizing energy consumption. Applying the MGO algorithm can assist companies or organizations in improving their operational efficiency, particularly in scheduling production processes involving no-wait permutation flows. Although MGO shows equivalent performance to the GWO algorithm, its practical implications remain significant, as it provides an effective alternative option in handling scheduling problems with a focus on energy efficiency.

5. Conclusion

This research successfully introduces modifying the mountain Gazelle optimizer (MGO) algorithm as an effective solution to the no-wait permutation flow shop scheduling problem, focusing on reducing energy consumption. Based on the convergence curve results, the convergence curve analysis results show that MGO can achieve convergence better than GWO, PSO, GA, COA, and FHO procedures. This result indicates that the proposed MGO procedure efficiently performs the optimal solution. Based on the average energy consumption test of algorithm performance results with Anova, the results show that MGO, in solving the no-wait permutation flow shop scheduling problem with the objective function, minimizes energy consumption in various solutions in Cases 1, 2, and 3. These results show that the MGO algorithm performs better than the GWO, PSO, GA, COA, and FHO algorithms in solving the no-wait permutation flow shop scheduling problem.

However, it should be recognized that the limitation of this study lies in the tightness of the variety of test data. Therefore, further exploration with more diverse datasets is required to strengthen the validity of the results. Furthermore, although MGO shows excellent performance, its comparison with other metaheuristic algorithms that have not been explored allows for a more comprehensive insight. Although this research successfully demonstrated the effectiveness of the MGO algorithm in handling the no-wait permutation flow shop scheduling problem with various case scenarios, some limitations still need to be noted. This research is limited to scheduling issues with a single objective: minimizing energy consumption. Further development could include research on multi-objective optimization that considers trade-offs between various criteria, such as production time and cost. In addition, testing MGO in a practical context in a real industrial environment will provide more valuable insights into its performance. For future research, it is also necessary to consider comparing MGO with other metaheuristic algorithms that this study has not explored. It will help determine the advantages and limitations of MGO in various cases. In addition, developing the MGO algorithm by incorporating other procedures that can enhance its ability to solve more complicated scheduling problems is a promising next step. By doing this, future research can further deepen our understanding of the potential of MGO in various optimization contexts and pursue more efficient and widely applicable solutions.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

For this research work, the authors contributed to Conceptualization, Utama, Sanjaya and Nugraha; methodology, Utama and Nugraha; software, Sanjaya; validation, Utama, Sanjaya, and Nugraha; formal analysis, Utama and Nugraha; investigation, Utama, Sanjaya, Nugraha; writing—original draft preparation, Sanjaya; writing—review and editing, Sanjaya and Utama; funding acquisition, Utama.

Acknowledgements

The authors would like to thank the reviewers and editors for their suggestions and input in this article. We would also like to thank the optimization laboratory of the Industrial Engineering department for allowing us to use its facilities.

References

[1] K. Fang, N. Uhan, F. Zhao, and J. W. Sutherland, "A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction", *Journal of*

Manufacturing Systems, Vol. 30, No. 4, pp. 234-240, 2011.

[2] D. M. Utama, A. A. P. Salim, and D. S. Widodo, "A novel hybrid archimedes optimization algorithm for energy-efficient hybrid flow shop scheduling", *International Journal of Advances in Intelligent Informatics*, Vol. 8, No. 2, 2022.

[3] D. M. Utama, M. D. Primayesti, S. Z. Umamy, B. M. N. Kholifa, and A. D. Yasa, "A systematic literature review on energy-efficient hybrid flow shop scheduling", *Cogent Engineering*, Vol. 10, No. 1, p. 2206074, 2023.

[4] J. Y. Ding, S. Song, and C. Wu, "Carbon-efficient scheduling of flow shops by multi-objective optimization", *European Journal of Operational Research*, Vol. 248, No. 3, pp. 758-771, 2016.

[5] D. M. Utama and C. Febrita, "Low-carbon no-idle permutation flow shop schedulling problem: giant trevally optimizer vs African vultures optimization algorithm", *Int J Adv Appl Sci,* Vol. 12, No. 3, pp. 195-204, 2023.

[6] Y. M. Risma and D. M. Utama, "AVOA and ALO Algorithm for Energy-Efficient No-Idle Permutation Flow Shop Scheduling Problem: A Comparison Study", *Jurnal Optimasi Sistem Industri*, Vol. 22, No. 2, pp. 126-141, 2023.

[7] S. A. Mansouri, E. Aktas, and U. Besikci, "Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption", *European Journal of Operational Research*, Vol. 248, No. 3, pp. 772-788, 2016.

[8] J. Grabowski and J. Pempera, "Sequencing of jobs in some production system", *European journal of operational research*, Vol. 125, No. 3, pp. 535-550, 2000.

[9] X. Wu and A. Che, "Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search", *Omega*, Vol. 94, p. 102117, 2020.

[10] F. Jolai, M. Rabiee, and H. Asefi, "A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times", *International Journal of Production Research*, Vol. 50, No. 24, pp. 7447-7466, 2012.

[11] K. Nailwal, D. Gupta, and K. Jeet, "Heuristics for no-wait flow shop scheduling problem", *International Journal of Industrial Engineering Computations*, Vol. 7, No. 4, pp. 671-680, 2016.

[12] M. T. Pereira and M. S. Nagano, "Hybrid metaheuristics for the integrated and detailed scheduling of production and delivery operations in no-wait flow shop systems", *Computers & Industrial Engineering*, Vol. 170, p. 108255, 2022.

[13] D. M. Utama, "An effective hybrid crow search algorithm for energy-efficient flow shop scheduling", In: *Proc. of AIP Conference Proceedings*, Vol. 2453, No. 1, p. 020040: AIP Publishing LLC, 2022.

[14] D. M. Utama, M. F. Ibrahim, D. S. Wijaya, D. S. Widodo, and M. D. Primayesti, "A Novel Hybrid Multi-Verse Optimizer Algorithm for Energy-Efficient Permutation Flow Shop Scheduling Problem", *Journal of Physics: Conference Series*, Vol. 2394, No. 1, p. 012006, 2022.

[15] P. D. Kusuma and A. Dinimaharawati, "Extended stochastic coati optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482-494, 2023, doi: 10.22266/ijies2023.0630.38.

[16] P. D. Kusuma, F. C. J. I. J. O. I. E. Hasibuan, and Systems, "Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour", *International Journal of Intelligent Engineering Systems,* vol. 16, no. 4, p. 264, 2023, doi: 10.22266/ijies2023.0831.22.

[17] P. D. Kusuma and A. L. Prasasti, "Walk-Spread Algorithm: A Fast and Superior Stochastic Optimization", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 5, pp. 275-288, 2023, doi: 10.22266/ijies2023.1031.24.

[18] P. D. Kusuma and A. Dinimaharawati, "Four Directed Search Algorithm: A New Optimization Method and Its Hyper Strategy Investigation", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 5, pp. 598-611, 2023, doi: 10.22266/ijies2023.1031.51.

[19] O. Engin and A. Güçlü, "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems", *Applied Soft Computing***,** Vol. 72, pp. 166-176, 2018.

[20] N. Khodadadi, E. S. M. E. Kenawy, F. D. Caso, A. H. Alharbi, D. S. Khafaga, and A. Nanni, "The Mountain Gazelle Optimizer for truss structures optimization", *Applied Computing and Intelligence*, Vol. 3, No. 2, pp. 116-144, 2023.

[21] K. Gao, Q. Pan, P. N. Suganthan, and J. Li, "Effective heuristics for the no-wait flow shop scheduling problem with total flow time minimization", *The International Journal of Advanced Manufacturing Technology*, Vol. 66, pp. 1563-1572, 2013.

Received: November 17, 2023. Revised: February 12, 2024.

[22] M. S. Nagano, H. H. Miyata, and D. C. Araújo, "A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times", *Journal of Manufacturing Systems*, Vol. 36, pp. 224-230, 2015.

[23] H. Ye, W. Li, A. Abedini, and B. Nault, "An effective and efficient heuristic for no-wait flow shop production to minimize total completion time", *Computers & Industrial Engineering*, Vol. 108, pp. 57-69, 2017.

[24] D. Laha and U. K. Chakraborty, "A constructive heuristic for minimizing makespan in no-wait flow shop scheduling", *The International Journal of Advanced Manufacturing Technology*, Vol. 41, pp. 97-109, 2009.

[25] Y. Wang, X. Li, R. Ruiz, and S. Sui, "An iterated greedy heuristic for mixed no-wait flowshop problems", *IEEE Transactions on Cybernetics*, Vol. 48, No. 5, pp. 1553-1566, 2018.

[26] S. W. Lin and K. C. Ying, "Minimizing makespan for solving the distributed no-wait flowshop scheduling problem", *Computers & Industrial Engineering*, Vol. 99, pp. 202-209, 2016.

[27] R. T. Moghaddam, A. R. R. Vahed, and A. H. Mirzaei, "Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm", *The International Journal of Advanced Manufacturing Technology*, Vol. 36, pp. 969-981, 2008.

[28] M. R. Santiago, S. Muthuswamy, P. Damodaran, and M. C. V. Gallego, "Scheduling batch processing machines in a no-wait flow shop using ACO to minimise the makespan", *International Journal of Industrial and Systems Engineering*, Vol. 27, No. 3, pp. 390-411, 2017.

[29] Q. Q. Zeng, J. Q. Li, R. H. Li, T. H. Huang, Y. Y. Han, and H. Y. Sang, "Improved NSGA-II for energy-efficient distributed no-wait flow-shop with sequence-dependent setup time", *Complex & Intelligent Systems*, Vol. 9, No. 1, pp. 825-849, 2023.

[30] F. Zhao, X. Hu, L. Wang, T. Xu, N. Zhu, and Jonrinaldi, "A reinforcement learning-driven brain storm optimisation algorithm for multi-objective energy-efficient distributed assembly no-wait flow shop scheduling problem", *International Journal of Production Research,* Vol. 61, No. 9, pp. 2854-2872, 2023.

[31] F. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem", *IEEE Transactions on Cybernetics*, Vol. 51, No. 11, pp. 5291-5303, 2021.

[32] F. Zhao, Z. Xu, X. Hu, T. Xu, and N. Zhu, "An improved iterative greedy athm for energy-efficient distributed assembly no-wait flow-shop scheduling problem", *Swarm and Evolutionary Computation*, Vol. 81, p. 101355, 2023.

[33] F. Zhao, T. Jiang, and L. Wang, "A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time", *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 7, pp. 8427-8440, 2023.

[34] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, "Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems", *Advances in Engineering Software*, Vol. 174, p. 103282, 2022.

[35] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[36] G. Venter and J. Sobieszczanski-Sobieski, "Particle swarm optimization", *AIAA Journal,* Vol. 41, No. 8, pp. 1583-1589, 2003.

[37] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms", *Search methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pp. 97-125, 2005.

[38] D. M. Utama, L. R. Ardiansyah, and A. K. Garside, "Penjadwalan Flow Shop untuk Meminimasi Total Tardiness Menggunakan Algoritma Cross Entropy–Algoritma Genetika", *Jurnal Optimasi Sistem Industri*, Vol. 18, No. 2, pp. 133-141, 2019.

[39] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems", *Knowledge-Based Systems*, Vol. 259, p. 110011, 2023.

[40] M. Azizi, S. Talatahari, and A. H. Gandomi, "Fire Hawk Optimizer: A novel metaheuristic algorithm", *Artificial Intelligence Review*, Vol. 56, No. 1, pp. 287-363, 2023.

[41] D. Yüksel, M. F. Taşgetiren, L. Kandiller, and L. Gao, "An energy-efficient bi-objective no-wait permutation flowshop scheduling problem to minimize total tardiness and total energy consumption", *Computers & Industrial Engineering*, Vol. 145, p. 106431, 2020.

[42] D. M. Utama, "Minimizing Number of Tardy Jobs in Flow Shop Scheduling Using A Hybrid Whale Optimization Algorithm", In: *Proc. of the International Conference on Industrial*

Received: November 17, 2023. Revised: February 12, 2024.

Automation, Smart Grid and its Application (ICIASGA) 2020, Madiun, Jawa Timur, Indonesia, Vol. 1845, No. 1, p. 012017, 2021.

[43] D. S. Widodo and D. M. Utama, "The Hybrid Ant Lion Optimization Flow Shop Scheduling Problem for Minimizing Completion Time", In: *Proc. of International Conference on Science and Technology 2019* Surabaya, Indonesia, Vol. 1569, No. 2, p. 022097, 2020.

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

[45] D. M. Utama, D. S. Widodo, M. F. Ibrahim, and S. K. Dewi, "An effective hybrid ant lion algorithm to minimize mean tardiness on permutation flow shop scheduling problem", *International Journal of Advances in Intelligent Informatics*, Vol. 6, No. 1, pp. 23-35, 2020.

[46] J. Carlier, "Ordonnancements a contraintes disjonctives", *RAIRO-Operations Research*, Vol. 12, No. 4, pp. 333-350, 1978.

[47] C. R. Reeves, "A genetic algorithm for flowshop sequencing", *Computers & Operations Research*, Vol. 22, No. 1, pp. 5-13, 1995.

[48] J. Heller, "Some numerical experiments for an M× J flow shop and its decision-theoretical aspects", *Operations Research*, Vol. 8, No. 2, pp. 178-184, 1960.