processes

Article

A Sequential Hybrid Optimization Algorithm (SHOA) to Solve the Hybrid Flow Shop Scheduling Problems to Minimize Carbon Footprint

M. Geetha [1,*], R. Chandra Guru Sekar [2], M. K. Marichelvam [3] and Ömür Tosun [4]

[1] Department of Mathematics, Kamaraj College of Engineering and Technology, S.P.G.C. Nagar, Vellakulam, Virudhunagar 625701, India
[2] Department of Mathematics, Mepco Schlenk Engineering College, Sivakasi 626005, India; chandragurusekar@mepcoeng.ac.in
[3] Department of Mechanical Engineering, Mepco Schlenk Engineering College, Sivakasi 626005, India; mkmarichelvam@mepcoeng.ac.in
[4] Department of Management Information Systems, Faculty of Applied Sciences, Akdeniz University, Antalya 07070, Turkey; omurtosun@akdeniz.edu.tr
* Correspondence: geethamkm1312@gmail.com

Abstract: In today's world, a situational awareness of sustainability is becoming increasingly important. Leaving a better world for future generations is becoming the main interest of many studies. It also puts pressure on managers to change production methods in most industries. Reducing carbon emissions in industry today is crucial to saving our planet. Theoretical research and practical industry requirements diverge, even though numerous researchers have tackled various strategies to handle carbon emission problems. Therefore, this work considers the carbon emission problem of the furniture manufacturing industry in Hosur, Tamilnadu, India. The case study company has a manufacturing system that resembles a hybrid flow shop (HFS) environment. As the HFS scheduling problems are NP-hard in nature, exact solution techniques could not be used to solve the problems. Hence, a sequential hybrid optimization algorithm (SHOA) has been developed in this paper to minimize the carbon footprint. In the SHOA, the pigeon-inspired optimization algorithm (PIOA) is hybridized sequentially with the firefly algorithm (FA). A computational experimental design is proposed to analyze the efficiency of the introduced strategy, and the solutions indicate that the developed approach could reduce the carbon footprint by up to 9.82%. The results motivate us to implement the proposed algorithm in the manufacturing industry to reduce the carbon footprint.

Keywords: carbon footprint; hybrid flow shop; scheduling; pigeon-inspired optimization algorithm (PIOA); firefly algorithm (FA)

Citation: Geetha, M.; Chandra Guru Sekar, R.; Marichelvam, M.K.; Tosun, Ö. A Sequential Hybrid Optimization Algorithm (SHOA) to Solve the Hybrid Flow Shop Scheduling Problems to Minimize Carbon Footprint. *Processes* **2024**, *12*, 143. https://doi.org/10.3390/pr12010143

Academic Editors: Michael C. Georgiadis and Olympia Roeva

Received: 3 December 2023
Revised: 1 January 2024
Accepted: 4 January 2024
Published: 6 January 2024

1. Introduction

The main objective of any manufacturing or service industry is to make a profit. The latest technological developments and the concepts of industrial engineering help these industries maximize their profits. But, in recent times, managers seem to understand that profit is not the only real goal to achieve. Leaving a positive impact on future generations in a sustainable environment also turns out to be important [1]. Ongoing usage of resources without thinking about their environmental effects becomes the main problem in most manufacturing companies. Thus, today the scenario has changed due to the universal environmental rules and regulations. Although there are different strategies to lessen the impacts on our world, better utilization of resources or reducing the waste of production facilities become the main research topics of the literature [2]. One of the proper ways to achieve these goals should be by designing a better schedule in a workshop environment.

It is our collective responsibility to reduce carbon emissions to mitigate the effects of greenhouse gas emissions, global warming, and climate change. Environment-friendly production also has a positive impact on consumers, which also increases their loyalty to the related companies. Therefore, most industries in the world attempt to lessen carbon emissions by adopting different strategies. In recent years, several researchers have attempted to reduce carbon emissions with the help of their potential research activities. Effective scheduling is an important method proposed by some shop floor engineers and operational analysts to minimize carbon emissions [3,4]. Scheduling always has a key role in the manufacturing environment and operations management. Overall, scheduling can be identified as the process of distributing existing inadequate resources in an effective manner to maximize or minimize certain objective functions [5]. Many scheduling environments are described in the literature, including flexible manufacturing system (FMS) scheduling, job shop scheduling, hybrid flow shop (HFS) scheduling, flow shop scheduling, parallel machine scheduling, single machine scheduling, and flexible job shop scheduling [6]. Among them, HFS scheduling is an important research area as many industries, such as textiles [7], furniture manufacturing [8], automobile component manufacturing [9], chemical industries [10], electronics industries [11], etc., resemble the HFS environment. The general layout of an HFS environment can be seen in Figure 1 [8].

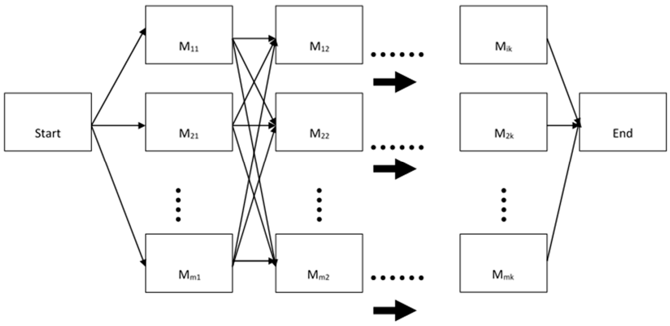

Figure 1. Layout of an HFS environment.

Arthanari and Ramamurthy [12] have addressed the HFS scheduling problems in their research. According to the research, the HFS scheduling problem is a type of combinatorial optimization problem that is NP-hard, or non-deterministic polynomial-time hard [13,14]. Therefore, exact solution techniques for these problems are difficult to obtain when the problem size ranges from moderate to large. Therefore, researchers use different types of heuristics and metaheuristics to solve HFS scheduling problems. Nowadays, scheduling researchers also concentrate on the reduction of carbon emissions. There is a discrepancy between the theoretical analysis and the actual requirements of industries, even though numerous scheduling researchers have addressed various scheduling problems with various goal functions [15,16]. Hence, in the present work, an effort is made to lessen the carbon footprint of a manufacturing industry located in Hosur, India. A sequential optimization solution methodology is developed to answer the related problem. The main contributions of the present work are:

- To develop a sequential hybrid optimization algorithm (SHOA) using the pigeon-inspired optimization algorithm (PIOA) and firefly algorithm (FA) to minimize the carbon footprint in an HFS environment.
- To solve the real industrial scheduling problem of a collaborative company using the SHOA.
- To develop random benchmark problem instances to evaluate the performance of the proposed SHOA and compare the results with other algorithms available in the literature.

The remainder of the research is structured as follows: The second section contains a thorough literature review on carbon emissions scheduling problems, the PIOA, and the FA. The problem definition is presented in Section 3. The solution methodology is described in Section 4. Computational experiments are discussed in Section 5. The concluding remarks are given in Section 6.

2. Literature Review

Here, the previous studies related to this paper are analyzed thoroughly. After reviewing the literature on the scheduling problems that handle carbon footprint minimization, the pigeon optimization algorithm and the firefly algorithm are addressed, respectively.

2.1. Scheduling Problems with Carbon Footprint Minimization Objective

There is a direct link between the energy consumption of a manufacturing facility and its environmental impact, which is mostly analyzed as a carbon footprint. Mouzon & Yildirim [17] developed a novel greedy randomized adaptive metaheuristic algorithm for multiple objectives to minimize the total energy consumption and total delay in a single machine environment. By minimizing energy consumption, they reduced their carbon footprint. Fang et al. [18] developed a novel mathematical approach to minimize the peak energy load, power consumption, and associated carbon footprint in a cast iron plate manufacturing plant. An enhanced genetic simulated annealing algorithm (GSAA) was used to reduce the makespan and energy consumption in a flexible flow shop environment [19]. A multiobjective ant colony optimization algorithm (MOACO) was presented by Luo et al. [20] for minimizing the makespan and electricity usage. A multi-objective genetic algorithm (MOGA) was investigated by Liu and Huang [21] to decrease the carbon footprint by minimizing the total weighted tardiness. In another multi-objective study, a teaching and learning-based optimization algorithm (MTLBOA) was proposed by Lin et al. [22] for finding the best process constraints in turning industry with makespan and also the carbon footprint.

Ding et al. [23] studied a hybrid model of the NEH heuristic and an iterated greedy algorithm to analyze the flow shop scheduling environment with numerous objectives. They developed a multiobjective solution strategy with makespan minimization and total carbon emission minimization in their work. They generated several random problem instances with numerous numbers of jobs and machines to validate the given methods. The computational solutions were measured against several algorithms given in the literature and proved to be better. Yan et al. [24] developed a multi-level optimization solution approach to obtain better results for the energy-efficient flexible flow shop scheduling environment under the makespan and total energy usage. After developing a mathematical model, the genetic algorithm (GA) was used to find the optimal schedules. The effectiveness of the given method was investigated using a case study. Lei et al. [25] introduced a teaching-learning-based optimization algorithm (TLBOA) for finding the minimum of the total energy usage and total tardiness in an HFS problem. The HFS scheduling problem was divided into three sub-problems, and a three-string coding method was used for the solution representation of these sub-problems. Several test instances were solved to analyze the efficiency of the given approach. Liu et al. [26] investigated the efficiency of a fruit fly optimization algorithm (FFOA) for minimizing the carbon footprints of all products and makespan in a flexible job shop environment using the data from a case problem. They identified product carbon footprints using the relationships between resources and products using these resources.

In their study, Zhang et al. [27] integrated the minimum values of total electricity usage and carbon footprint into the objective function of their algorithm. Additionally, they incorporated the utilization of total energy and carbon emissions as constraints in their proposed approach. The researchers put forth the use of integer programming as a means to solve the mathematical model. Lu et al. [28] designed a multi-objective approach for the multi-stage HFS problems under makespan, noise pollution, and overall energy consumption. An improved gray-wolf algorithm was used to solve the problems. Nasiri et al. [29] used an integer programming method to reduce the sum of weighted tardiness and energy usage in an HFS environment. Pan et al. [30] developed an effective imperialist competitive algorithm (ICA) for multiobjective, low-carbon parallel machine scheduling. Piroozfard et al. [31] analyzed the minimization of the sum of the carbon footprint and total late work indicators in a flexible job shop environment. To achieve this objective, they employed a MOGA. The authors indicated that the choice of these two roles was indicative of aspirations rooted in sustainability and classical principles.

Wu et al. [32] analyzed a multi-criterion, flexible flow shop scheduling environment considering makespan and low-carbon scheduling objectives. The carbon emissions were measured with the use of renewable and non-renewable energy on a machine. They proposed a GA-based solution in their work. Meng et al. [33] analyzed the HFS problem to minimize the overall energy consumption under makespan and energy usage that can be varied under machines with different turning on/off state constraints. GA was used for solving the integrated mathematical model. Zhou and Liu [34] analyzed the effect of environmental pollution and increasing energy costs within an HFS with fuzzy processing times. Total weighted delivery time and energy usage were the objective functions. Chen et al. [35] have attempted to decrease the energy expenditure and minimize the completion time of a production system. They developed a multi-objective mixed-integer modeling approach to achieve energy efficiency HFS with lot streaming for the minimizing of the sum of two objectives, which are the makespan of the production and electric power usage.

Wang et al. [36] investigated a real-life scenario of a glass manufacturing business. A two-stage HFS in which machine eligibility in the first phase and batch machine in the following phase were selected. Similar to previous studies, makespan and total energy usage (which is preferred to reflect energy-efficient manufacturing) were used to measure the process. They considered time-of-use electricity prices and machine states (given as online, off, and idle) in their research. Different local search-based metaheuristics were used. The study conducted by Cai and Lei [37] addressed the management of a distributed energy-efficient hybrid flow shop scheduling environment that incorporates fuzzy processing time. To optimize the scheduling process, they introduced a cooperative shuffled frog-leaping algorithm (CSFLA). The objective of the research was to simultaneously determine the optimal values for fuzzy makespan, total agreement index, and fuzzy total energy usage. Shi et al. [38] studied a sustainable HFS environment under consideration of processing time, energy consumption, and carbon emissions. Using GA, dynamic scheduling was analyzed. Wang and Wang [39] investigated the energy-focused distributed HFS, considering makespan and energy usage. Using a multi-objective approach, a hybridized cooperative memetic algorithm (HCMA) was proposed to obtain optimal solutions. Zuo et al. [40] addressed the energy-efficient HFS scheduling problems considering the objectives of makespan, total tardiness, and total energy usage. With these objectives, the green scheduling and sustainable manufacturing needs of policymakers were achieved. Under variable speed constraints, an artificial bee colony algorithm (ABCA)-based solution methodology was developed.

2.2. Literature Review on Pigeon-Inspired Optimization Algorithm (PIOA)

The pigeon-inspired optimization algorithm (PIOA) is a metaheuristic based on swarm intelligence that was developed by Duan and Qiao [41]. The algorithm imitates the natural action pattern of pigeons in nature. While moving within a swarm, each member of this swarm (pigeons) has their own position, velocity, and personal best position; both of these are used for understanding their moves in a search space. The search activity of pigeons can be analyzed in two different phases: the first is based on their movements, while the second is related to the other pigeons in the swarm. The simple PIOA is described in [42–44].

Goel [45] used PIOA to search for the shortest path from a given point and measured the efficiency of the approach using Dijkstra's algorithm. Hao et al. [46] enhanced the basic PIOA to solve the unmanned aerial vehicle (UAV) assignment problem under energy consumption. Sun and Duan [47] modified the PIOA with the prey–predator strategy to prevent the algorithm from trapping into a locally optimal solution. Their proposed approach was used for protein–protein interaction (PIP) parameter adjustment. Zhang and Duan [48] suggested a predator–prey pigeon-inspired optimization (PPPIO) algorithm to solve the three-dimensional path planning problem of UAVs. The authors put forth the concept of predator–prey dynamics as a means to improve the overall characteristics of global optimization and accelerate the convergence rate. The researchers demonstrated that the performance of the PPPIO technique outperformed both the PIOA and particle swarm optimization (PSO) algorithms. Deng et al. [49] integrated the PIOA with membrane computing to address the parameter design challenges encountered in an industrial motor.

According to the findings of Hu et al. [50], the utilization of PIOA in certain intricate settings results in the attainment of local optima, a sluggish rate of convergence, and unstable solution attributes. Therefore, to deal with these shortcomings, they applied an adaptive weighted approach to PIOA and analyzed the UAV route planning problem. Pei et al. [51] combined PIOA with the quantum chaotic process to find the optimal fuzzy control strategy for a hybrid electric vehicle. Rehman et al. [52] hybridized the PIOA with the GA to minimize electricity costs while minimizing user discomfort. Liu et al. [53] developed an improved PIOA to cope with a nonlinear optimization problem. In the improved IPIOA, the authors introduced PSO, an inverse factor, and a Gaussian factor, and better results were obtained. Shang et al. [54] addressed a multi-objective PIO (MOPIO) approach to solve a community detection problem in network science with negative ratio associations and ratio cuts. They proposed a crossover strategy to enhance the solution quality. They compared the quality of results from their algorithm with other multi-objective algorithms studied in previous papers from the literature and concluded that the MOPIO solution methodology provided better results. Modified and improved versions of binary PIOA were studied by the researchers to solve optimization problems in different fields [55,56]. The researchers have created enhanced iterations of PIOA to address the challenges posed by the dynamic facility layout problem [57] and the 0–1 knapsack problem [58]. An oppositional PIOA was suggested in [59] to solve the economic load dispatching problems. There are only a few studies in which PIOA is applied to scheduling problems. Fu et al. [60] studied a fuzzy production environment under maintenance. The MOPIO algorithm was developed using fuzzy makespan as the secondary objective. Wu et al. [61] applied PIOA to address the flexible job shop scheduling problem (FJSP) to minimize the makespan. Lei et al. [62] proposed a hybrid bat and pigeon for solving autonomous vehicle navigation and mapping. Ding and Dong [63] developed an improved PIOA to solve the continuous function optimization problem. Recently, Torky et al. [64] solved the financial crisis problems using the PIOA.

2.3. Literature Review on Firefly Algorithm (FA)

The recent versions of FA and their applications were addressed in [65]. The study conducted by Marichelvam et al. [66] examined the HFS scheduling environment, specifically focusing on analyzing the makespan and mean flow time using the FA. Karthikeyan et al. [67] proposed a solution methodology for addressing the challenges posed by multi-objective flexible job shop scheduling problems. The integration of local search techniques with FA effectively manages the maximum completion time, workload of the crucial machine, and total workload. In Fan et al. [68], a two-stage HFS environment with part arrivals and on-time delivery rate criteria was solved with FA. Marichelvam and Geetha [69] used FA to minimize the weighted sum of makespan, total flow time, and machine idle time in a multi-stage environment. Chakaravarthy et al. [70] analyzed the performance of the FA against the AIS (artificial immune system) algorithm for makespan and total flow time criteria in the m-machine flow shop scheduling problem. It is a given that both algorithms give better solution quality than other well-known solution methodologies.

Marichelvam and Geetha [71] investigated total flow time minimization for the m-machine flow shop scheduling problems. Qamhan et al. [72] integrated periodic maintenance, setup times, and release dates with makespan in a real-life problem. A hybridized FA was used for the solution. A chaos-based FA was proposed by Lo et al. [73] to solve the permutation flow shop scheduling problems under makespan. Kaya et al. [74] improved the efficiency of FA with chaos theory and local search strategies for makespan in a flow shop scheduling environment. Rashid and Osman [75] studied the energy-efficient HFS under makespan and energy utilization. Comparing FA against other well-known metaheuristics shows the superiority of the developed FA in the problem. The details of the FA to solve various applications can be seen in [76,77]. Ghasemi et al. [78] developed a new version of FA with improved global exploration for optimization problems in engineering. Bacanin et al. [79] solved the feature selection problem using the FA. Ezzeldin et al. [80] hybridized FA with GA and a particle swarm optimization algorithm to solve water distribution network problems. Sheeba and Uma Maheswari [81] proposed an enhanced FA for cloud computing applications. Recently, Villaruz et al. [82] suggested a scouting FA to solve global optimization problems.

From the above literature review, it is evident that there is a potential scope to address the hybrid flow shop scheduling problems to minimize the carbon footprint. Also, it is observed that the application of PIOA to solve scheduling problems is limited in the literature. Hence, in the present work, an attempt is made to solve the hybrid flow shop scheduling problems to minimize the carbon footprint. The problem definition is described in the next section.

3. Problem Definition

The primary objective of this paper is to reduce the carbon footprint in a hybrid flow shop (HFS) environment. General scheduling approaches have been studied a lot in the literature, but integrating them with energy awareness is a recent topic. The industry that resembles the HFS environment is equipped with different types of machines for various processes. These machines consume electrical power. The generation of this electricity produces carbon emissions. Hence, if an industry minimizes its completion time (makespan), it minimizes carbon emissions. The makespan in a scheduling context can be defined as the time at which the final job completes its manufacturing process [5]. The HFS scheduling problem is the combination of M stages arranged in a sequence. Every stage, denoted by k, where k ranges from 1 to M, consists of a parallel combination of m_k identical machines. There is a collection of jobs i, where i = 1, 2,..., N, that must be executed on any available machine at any given stage. The processing times for job j at different stages are denoted as P_{1i}, P_{2i}, . . ., P_{ki}. It is worth noting that the processing time may be zero for certain jobs, as there are instances where certain tasks do not need to undergo operations at specific stages. The mathematical modeling to minimize the carbon footprint is described below.

Mathematical Formulation

The following assumptions are integrated into the mathematical model:

1. The number of stages and the number of machines are known.
2. The number and the operating times of each job in the manufacturing process are both known in advance and cannot be changed.
3. All the tasks involved in the production process are available at the first stage at time zero.
4. The act of preemption is prohibited.
5. For each job, their setup and transportation times are added to the operation times.
6. Each machine can only process a single job at any given time.
7. All machines are fully operational during the entire scheduled time frame (maintenance activities, i.e., machine breakdowns, are not taken into account).
8. Only electrical power is used on the shop floor for the operation of machines.

The objective function is to minimize the carbon footprint by minimizing the makespan.

$$Minimize\ C_{fp} \tag{1}$$

Equation (1) is used to minimize the carbon footprint by minimizing the makespan. According to the above assumptions, the mathematical basis of the energy-efficient problem can be described as described by Paternina-Arboleda et al. [83]:

$$Minimize\ C_{max} \tag{2}$$

Subject to:

$$C_{max} \geq C_{is}, \text{ for all } s = 1, 2, \ldots, M, j = 1, 2, \ldots, N \tag{3}$$

$$C_{is} = S_{is} + P_{si} \tag{4}$$

$$\sum_{j=1}^{m} Y_{ijs} = 1 \text{ forall } s = 1,\ 2, \ldots,\ M,\ i = 1,\ 2, \ldots,\ N \tag{5}$$

$$C_{is} \leq S_{i(s+1),} \text{ for } s = 1, 2, \ldots, M - 1 \tag{6}$$

$$S_{hs} \geq C_{is} - BW_{his}, \text{ for all given job pairs } (h, i) \tag{7}$$

$$S_{is} \geq C_{hs} + B - 1, \text{ for all given job pairs } (h, i) \tag{8}$$

$$S_{i1} \geq R_i \text{ for all } i = 1, 2, \ldots, N \tag{9}$$

$Y_{ijs} \in \{0, 1\}$, $W_{his} \in \{0, 1\}$,
for all given $i = 1, \ldots, N$

$$j = 1, 2, \ldots, m_s, \text{ and } s = 1, 2, \ldots, M \tag{10}$$

$$C_{is} \geq 0, \text{ for all } s = 1, 2, \ldots, M, i = 1, 2, \ldots, N \tag{11}$$

The problem addressed in this study is about reducing the carbon footprint by achieving a better job schedule with lower energy consumption by minimizing the makespan. This is given in Equation (2). Equation (3) ensures that the makespan is equal to or greater than the completion time of the last job. The optimality conditions are always fulfilled if C_{max} is positive according to the objective function when the effort is minimized. Equation (4) measures the completion time of job i at stage s. The assignment of each work to a single machine in each stage is ensured by Equation (5), while Equation (6) allows the starting point of each job only after it has been completed in the previous stage. Using Equations (6) and (7) in conjunction ensures that the mathematical model guarantees that only a single job can be assigned to a machine within a given stage at any given time. If W_{his} equals 1 and job h was completed before job i, Equation (7) is easy to fulfill. Then Equation (8) guarantees that the start of job i in stage s depends on the completion of job h. If $W_{his} = 0$ means that job i precedes job h, then it can be observed that Equation (8) is easily satisfied, and the starting time of job h at stage s must be equal to the completion time of job i at stage s to fulfill Equation (7). Then, Equation (9) imposes a constraint on the start of job scheduling, which requires that the start times of a job must be after the release times of jobs in the system. Furthermore, Equation (10) enforces binary values of zero or 1 for both variables Y_{ijs} and W_{his}. Equation (11) represents the non-negative constraint.

4. Sequential Hybrid Optimization Algorithm (SHOA)

In this present work, two different metaheuristics are combined serially. The pigeon-inspired optimization algorithm (PIOA) is hybridized sequentially with the firefly algorithm (FA). In this hybrid strategy of SHOA, the results from the PIOA are taken as initial solutions to the FA. The steps of the hybrid approach are illustrated below.

4.1. Pigeon-Inspired Optimization Algorithm (PIOA)

Mimicking the flying patterns of the pigeons, including their behaviors and the intelligence of a pigeon swarm for efficient food searching, is the main starting point of the PIOA. Long-distance flying and finding their route back to the starting point (or pigeon's home) is one of the astonishing points of the pigeons. Various indicators are employed in this procedure, namely the solar position, the terrestrial magnetic field, and the optical prominence of geographical features. The main communication and navigation are based on a leader pigeon. By communicating between the flock and leader, pigeons preserve their side-by-side flocking distance. In the PIOA, to model this homing trait, two operators, known as landmark operators and map and compass operators, are used.

4.1.1. Map and Compass Operator

A simple phenomenon called magneto reception is one of the main advantages of pigeons in helping to create a map using the help of Earth's magnetic field. Secondly, the sun's altitude also helps them adjust their position to their desired destination, like a compass. But the dependency of these two helping instincts lessens as they move closer to the destination (or hometown).

There are two important pieces of information we need to know to decide the further movement of a pigeon in the swarm. These are its initial position and current velocity.

In a search space with N-dimensions, $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,N})$ indicates the position of the ith individual in the swarm. Similarly, to represent the location change of a pigeon, its velocity given as $V_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,N})$ can be used. Here i = 1, 2, ..., M represents the size of the group. And finally, $G = (g_1, g_2, \ldots, g_N)$ is the global best position of the swarm determined by analyzing all positions in the swarm.

The updated location and velocity of pigeon i at the tth iteration can be calculated as follows:

$$v_{i,j}^{t} = v_{i,j}^{t-1} e^{-Rt} + \rho\left(g_j^{t-1} - x_{i,j}^{t-1}\right) \tag{12}$$

$$x_{i,j}^{t} = x_{i,j}^{t-1} + v_{i,j}^{t} \tag{13}$$

The velocity of a pigeon is determined by Equation (12), which takes into account the previous velocity of the pigeon and the distance to the global best location of the group. The operator labeled R is responsible for controlling the influence of recent velocity information, while ρ represents a random number uniformly distributed over the range [0, 1). The pigeon is able to adjust its position by using Equation (13) after changing its velocity.

4.1.2. Landmark Operator

Similar to how we remember our surroundings to find our way, pigeons use landmarks or already-known patterns to determine their route to their destination. If a similarity occurs, the pigeons can find their way quickly and fly directly to the appropriate destination. Otherwise, it is best to save energy by following the pigeons that already have information about the landmarks. Based on the fitness value of each individual, the number of swarms can be halved for each generation. The majority of the swarm tends to align with the central group, while the minority aligns with the superior subgroup. This assumption is primarily based on the fact that, within a swarm, the central pigeon is considered to be correctly oriented.

$$c_j^t = \frac{\sum_{i=1}^{N}\left(x_{i,j}^{t-1} f\left(\left(x_i^{t-1}\right)\right)\right)}{N\sum_{i=1}^{N}\left(f\left(x_i^{t-1}\right)\right)} \tag{14}$$

Equation (14) is used to select the middle pigeon. Where i = 1, 2, ..., N, and N is the size of the better part. $F(x)$ is the fitness value of the ith pigeon at the tth iteration. Equation (15) can be used to update the position of a pigeon.

$$x_{i,j}^{t} = x_{i,j}^{t-1} + r\left(c_j^t - x_i^{t-1}\right) \tag{15}$$

Using a two-step approach, first by employing a map and compass operator and then a landmark operator, improves the search efficiency of PIOA by balancing exploration and exploitation.

4.2. Firefly Algorithm

The second part of the hybrid strategy uses a bio-inspired algorithm known as the firefly algorithm. The firefly algorithm is derived from the collective behavior of fireflies in their social interactions. Most firefly species emit short and uniform flashes of light in their natural habitat. The configuration of the flashes has different characteristics for each individual species. In general, the bioluminescent flash emitted by fireflies serves as a signal to attract potential mates and prey. The flashes also serve as a defensive warning tool for the individuals. The basis of this algorithm has relied on the following guidelines developed by Yang and He [58] that help us convert swarm intelligence into an optimization approach.

1. All firefly species exhibit unisexuality, allowing any firefly to be attracted to another firefly regardless of their gender.
2. The attractiveness factor is influenced by the level of light intensity. Therefore, while comparing two unique fireflies, the one that emits a lower level of brightness will be attracted to the one that emits a higher level of brightness. Also, attractiveness is affected by distance, and for both flies, it will decrease as the distance between them increases. For a given specific firefly, it will move randomly if there are no brighter fireflies.
3. The brightness of a given species can be analyzed or measured with the shape of the objective value in a d-dimensional environment. For the maximization effort, the brightness is generally related to the objective function value. If the goal is to minimize, then brightness can be the inverse of the objective value.

The movement of a firefly is based on the attractiveness of the individuals in the swarm and the distance between them. The attractiveness can be calculated by the light intensity of a firefly, which varies with distance. It is given as:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (16)$$

Here, γ represents the light absorption constant. The measurement of the spatial distance between two given firefly species can be calculated as follows:

$$r_{kl} = \|X_k - X_l\| = \sqrt{\sum_{k=1}^{d} (X_{k,o} - X_{l,o})^2} \quad (17)$$

The combination of these two equations therefore results in the movement of a firefly:

$$X_k = X_k + \beta_o e^{-\gamma r_{kl}^2} (X_l - X_k) + \propto \left(rand - \frac{1}{2} \right) \quad (18)$$

4.3. The Steps in SHOA

The various steps in the development of SHOA are shown below. The flow chart of the SHOA is shown in Figure 2.

1. The parameters of both the PIOA and FA are defined. For the PIOA, the dimensions of the pigeon group, the factor associated with the map and compass, the maximum number of iterations of the map and compass operator, and the maximum number of iterations of the landmark operator are the control parameters used for performance evaluation. The attractiveness of a firefly β_0, the light absorption coefficient γ, and the randomization parameter α, and the number of iterations N are the parameters of FA. The parameters used in [41] are used for the PIOA. The parameters available in the literature [66] are used in the present work for the FA.
2. In a simple PIOA, the preliminary solutions are produced randomly. In this study, to improve the quality of the solution, a preliminary solution was generated using the NEH heuristics method as described by Nawaz et al. [84]. Then, we generated the remaining solutions arbitrarily. A uniform random number generator generates the continuous positional values randomly within the range of 0 and 1.
3. Originally, the PIOA's primary objective was to address optimization issues of a continuous nature. Hence, it is evident that the current version of PIOA is not suitable for directly addressing optimization issues that possess discrete characteristics. Bean [85] came up with the smallest position value (SPV) heuristic, which is used in this work to make continuous PIOA work for discrete flow shop scheduling problems.
4. For each pigeon in the swarm, measure their objective function values. Based on these values, the best pigeon of the swarm, X_{gbest}, can be selected.
5. Then, using the map and compass operations, update the velocity and position velocity of each species in the swarm.
6. The objective function of all pigeons is evaluated, and the best pigeon, X_{gbest}, is determined.

7. Steps 4 and 5 are iteratively executed until the designated number of iterations, denoted as N_1, is reached.
8. The landmark operator is executed to update the velocity and position of each pigeon.
9. Iteratively repeat Step 7 until reaching the specified number of iterations, denoted as N_2.
10. By the end of step 8, the best solution for the PIOA is obtained.
11. This solution is given as input to the Firefly algorithm. Step 10 provides the best solution, and in addition, some more solutions are randomly generated.
12. For these solutions (fireflies), the light intensity (objective function value) is determined.
13. Using the light absorption coefficient, the attractiveness of fireflies is calculated.
14. With the help of attractiveness, the movement of the fireflies is updated (the solutions are updated).
15. Steps 12 to 14 are continuously repeated until the previously determined number of generations is achieved.
16. Print the best result.

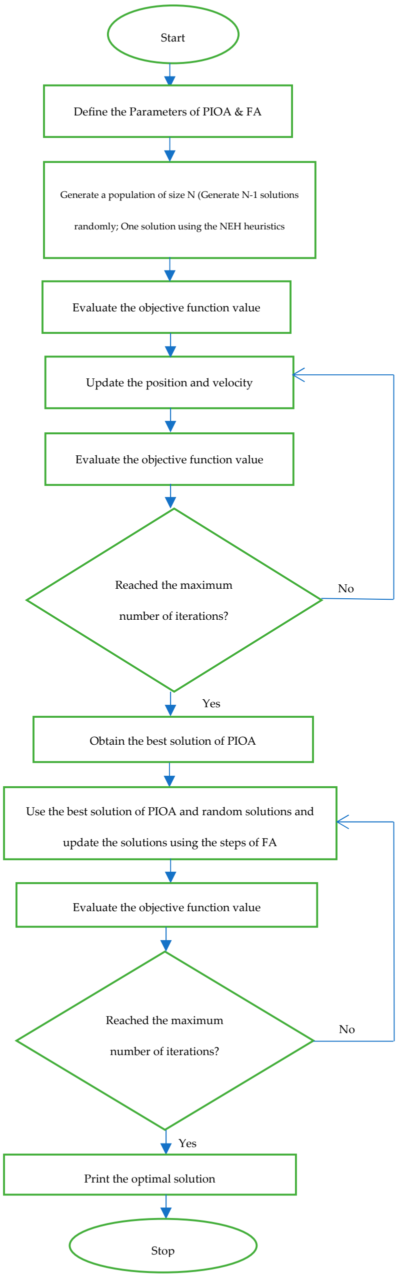

Figure 2. Flowchart of the SHOA.

4.4. Solution Representation

Proper design of solution representation plays a vital role in the PIOA as well as the FA. The search space consists of an n-dimensional region that includes n jobs. Therefore, each dimension corresponds to a job. The solution vector used for the algorithm gives the continuous position values of pigeons in the solution space. The SPV heuristic changes the continuous position values of the pigeons to the discrete job permutation. An example solution for a pigeon with six jobs is given in Table 1.

Table 1. Solution example for job positions.

	Dimension j					
	1	**2**	**3**	**4**	**5**	**6**
y_{ij}	0.66	0.18	0.79	0.30	0.58	0.86
jobs	4	1	5	2	3	6

5. Computational Experiments

Two types of experiments are conducted to validate and measure the performance of the designed hybrid strategy. Initially, a real industrial scheduling problem in the furniture manufacturing industry is considered. Later, we solve and compare some random benchmark problems against other metaheuristics from the literature.

5.1. Data from a Furniture Manufacturing Industry

This study uses the scheduling problem of a furniture manufacturing industry located in Hosur, India. The furniture manufacturing company produces several products. The study focuses on studying the process of making five-drawer vertical media storage cabinets among the different product types. The five-drawer flat media storage cabinets comprise 20 different parts. Each part is to be considered a job. The parts are manufactured in lots. The facility has a lot size of 600. The manufacturing process includes five different stages, such as punching, bending, welding, power pressing, and drilling. Each stage is made of several identical parallel machines. The different stages, the number of machines available in each stage, and the power consumption can be seen in Table 2. The processing time for each job is given in Table 3.

Table 2. The details of stages and number of machines for the case study.

Stage No.	**Stage Name**	**Number of Machines**	**Power Requirements (in kW)**
I	Punching	6	7.50
II	Bending	8	1.50
III	Welding	6	4.50
IV	Power pressing	3	3.75
V	Drilling	1	1.00

Table 3. Processing times for the jobs (in seconds).

Jobs→ Stages↓	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
I	60	36	60	60	24	60	30	18	48	60	0	0	48	0	0	72	0	0	0	0
II	48	64	24	0	32	16	0	0	72	24	0	0	80	0	0	56	0	0	0	0
III	0	12	60	0	0	72	0	0	60	24	0	0	0	0	0	0	0	0	0	0
IV	0	0	0	0	0	0	0	72	0	0	90	30	0	60	30	0	75	60	48	72
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0

The company does not apply any algorithms to schedule the jobs. For the management of the job orders, they follow the first-in-first-out (FIFO) dispatching rule only. The makespan is the total completion time of all the jobs in the production system. At the initial phase, the makespan for the FIFO is 145,800 s, and for completing all the jobs, 415.375 kWh of power is required by the furniture manufacturing company. The Central Electricity Authority (CEA) of India states that commercial electrical energy emits 0.85 kg of CO2 per 1 kWh [86]. The furniture manufacturing company emits a total of 353.07 kg of carbon by adopting the company sequence (CS) for manufacturing the five-drawer vertical media storage cabinet.

Implementing the SHOA yields an optimal schedule. By using the optimal schedule, it is possible to reduce the job completion cycle, thus the makespan, and therefore the total commercial electrical energy demanded by the furniture manufacturing company. Figure 3 shows the carbon footprint of the furniture manufacturing company before and after implementing the proposed algorithm. From the results, it is evident that the proposed SHOA reduced the carbon footprint by nearly 9%. This 9% minimization would reduce the environmental impact significantly. To validate the proposed algorithm, we have decided to develop random benchmark instances. In the next section, we will explain the details.

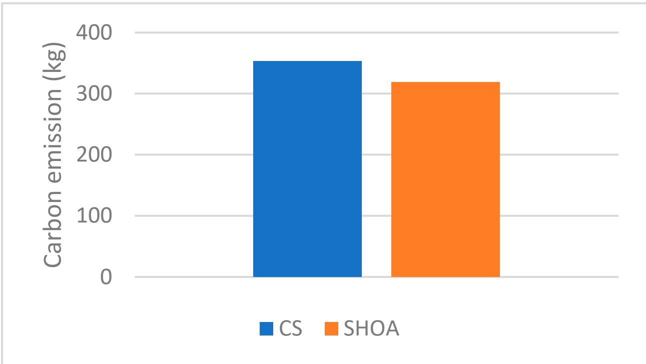

Figure 3. The carbon footprint of the furniture manufacturing company.

5.2. Experimental Simulation of the Algorithm

To further analyze the efficiency of the given SHOA, a second experimental design was evaluated by conducting a simulation with randomly generated problems, including different parameter settings. Table 4 presents the parameter values used in the experimental analysis.

Table 4. Parameter values for the benchmark problems.

Name of the Parameter	Values
Number of jobs	100
Number of stages	2, 5, 10
Number of machines at each stage	2, 5, 10
Statistical distribution to generate processing times	U (0, 100)
The power rating of machines (kW)	U (1, 100)
The attractiveness of the species in FA (β_0)	0, 0.50 and 1.00
Coefficient of light absorption, γ	0.50, 0.75 and 1.00
Randomization factor, α	0, 0.50 and 1.00
Max. number of iterations for the map and compass operator	100
Max. number of iterations for landmark operator	200
Max. number of generations of FA	500

We conducted $1 \times 3 \times 3 \times 1 \times 1 \times 3 \times 3 \times 3 \times 1 \times 1 \times 1 = 243$ experiments to measure and compare the performance of the developed hybrid strategy. Each problem instance is repeated for 20 runs, and the average carbon footprint values are considered. The performance of the proposed SHOA is measured against other metaheuristics from the literature, such as ACO [20], ABCA [40], CSFLA [37], FA [65], GA [24], HCMA [39], ICA [30], simple PIOA [41], and TLBOA [25]. The relative deviation index (RDI) is the performance indicator used to compare the algorithm's efficiency. Equation (19) calculates the RDI.

$$\mathrm{RDI} = \frac{C_{fp-Algorithms} - C_{fp-optimal}}{C_{fp-optimal}} \times 100 \tag{19}$$

where,

$C_{fp-Algorithms}$ = Carbon footprint value found by different algorithms
$C_{fp-optimal}$ = optimal (minimum) carbon footprint value obtained

Equation (20) is used to calculate the mean relative deviation (MRDI) index values from the RDI values.

$$\mathrm{MRDI} = \sum_{P=1}^{243} \mathrm{RDI}/243 \tag{20}$$

Table 5 presents the MRDI comparison of different metaheuristics. It is observed that the performance of the proposed SHOA is better than that of ACO [20]. From the table, it can also be seen that the SHOA provides a 5.32% improvement concerning ABCA [40] and 4.64% concerning CSFLA [37]. The SHOA provides better results than the simple PIOA [41] and FA [65] by 2.68% and 2.42%, respectively. This is because of the hybridization of both the PIOA and FA. The SHOA results are also better than those of other algorithms like GA [24], HCMA [39], ICA [30], and TLBOA [25]. The SHOA provides better results for both the industrial scheduling problems and the random benchmark problems. Hence, the proposed algorithm could be used for other industrial scheduling problems to minimize the carbon footprint, which would reduce the environmental impact significantly.

Table 5. MRDI comparison of different algorithms.

Sl. No.	Algorithms	MRDI
1	ACO	3.16
2	ABC	5.32
3	CSFLA	4.64
4	FA	2.42
5	GA	7.42
6	HCMA	9.42
7	ICA	7.91
8	PIOA	2.68
9	SHOA (present work)	0.00
10	TLBOA	5.46

6. Conclusions

Scheduling refers to the systematic allocation of finite resources within a production setting to execute various tasks to optimize specific goal functions. In the past, production costs or completion time-based objectives were important, but nowadays sustainability or energy-efficient solutions gain more attractiveness apart from them for a better future. Therefore, combining those different objectives is the starting point of the study. One of the proper ways to ensure a sustainable manufacturing environment is by reducing waste in the production process. This can be achieved with better usage of resources or a transformation to better technologies. This study presents a sequential hybrid optimization approach for addressing a practical scheduling problem in the furniture manufacturing business to minimize the carbon footprint. Following the development of a mathematical model, the firefly method is integrated with the pigeon-inspired optimization technique to address and improve the quality of the solution. The findings indicate that the use of the suggested algorithm has the potential to achieve a reduction of up to 9.82% in the carbon footprint. The efficacy of the algorithm under consideration was further confirmed through its application to benchmark problems. The results of the proposed algorithm were measured against different metaheuristics described in the literature. The outcomes achieved through the implementation of the suggested algorithm demonstrate a higher level of performance compared to alternative methods. Although the proposed method achieved improved outcomes, certain limitations were identified in this study. A lot of assumptions are considered in the current work, which may be practically insignificant. The removal of these assumptions would allow for the validation of the performance of the suggested method. Consideration of scheduling uncertainties such as rush orders, canceled orders, absenteeism of workers, and learning and forgetting effects of workers would be another interesting future scope of the present work. The algorithm under consideration has potential applications in lot streaming hybrid flow shop scheduling problems, and the reduction in the carbon footprint may be investigated. The proposed algorithm would be used to solve the carbon footprint of hybrid flow shop scheduling problems with multiple objectives.

Author Contributions: Conceptualization, M.G. and M.K.M.; methodology, R.C.G.S.; software, M.K.M.; validation, M.G. and R.C.G.S.; formal analysis, Ö.T.; data curation, Ö.T.; writing—original draft preparation, M.G.; writing—review and editing, M.K.M.; supervision, R.C.G.S.; project administration, Ö.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

The following representations are used in the mathematical model:

C_{hs}	The duration required to complete job h at stage s
C_{iM}	The duration required to complete the job i at the stage M
C_{is}	The duration required to complete the job i at stage s
C_{fp}	Carbon footprint
C_{max}	Makespan
N	Number of jobs to be scheduled (index i)
P_{si}	Operation time of the job i at stage s
R_i	Ready time of the job i
B	A consistent and unchanging value or quantity ($B \rightarrow \infty$)
M	The quantity of manufacturing stages (index s)
m_s	The number of machines that exhibit similarity at a certain stage, denoted as s
S_{hs}	The commencement time for a certain task, denoted as h, at a particular stage, denoted as s
S_{is}	The commencement time for a certain task, denoted as i, within a particular stage, referred to as s
S_{i1}	The commencement time for task i during the first stage
W_{his}	The binary variable takes the value of 1 when task h is scheduled before job i during processing at stage s, and 0 otherwise
Y_{ijs}	The binary variable takes the value of 1 when job i is allocated to machine j during step s, and 0 otherwise

References

1. Assia, S.; El Abbassi, I.; El Barkany, A.; Darcherif, M.; El Biyaali, A. Green scheduling of jobs and flexible periods of maintenance in a two-machine flowshop to minimize makespan, a measure of service level and total energy consumption. *Adv. Oper. Res.* **2020**, *2020*, 9732563. [CrossRef]
2. Alvarez-Meaza, I.; Zarrabeitia-Bilbao, E.; Rio-Belver, R.M.; Garechana-Anacabe, G. Green scheduling to achieve green manufacturing: Pursuing a research agenda by mapping science. *Technol. Soc.* **2021**, *67*, 101758. [CrossRef]
3. Hidri, L.; Alqahtani, A.; Gazdar, A.; Ben Youssef, B. Green scheduling of identical parallel machines with release date, delivery time and no-idle machine constraints. *Sustainability* **2021**, *13*, 9277. [CrossRef]
4. Li, Y.Z.; Pan, Q.K.; Gao, K.Z.; Tasgetiren, M.F.; Zhang, B.; Li, J.Q. A green scheduling algorithm for the distributed flowshop problem. *Appl. Soft Comput.* **2021**, *109*, 107526. [CrossRef]
5. Baker, K.R.; Trietsch, D. *Principles of Sequencing and Scheduling*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013; pp. 1–3.
6. Pinedo, M.L. *Scheduling: Theory, Algorithms, and Systems*, 4th ed.; Springer: New York, NY, USA, 2012; pp. 13–33.
7. Grabowski, J.; Pempera, J. Sequencing of jobs in some production system. *Eur. J. Oper. Res.* **2000**, *125*, 535–550. [CrossRef]
8. Marichelvam, M.K.; Prabaharan, T.; Yang, X.S. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Appl. Soft Comput.* **2014**, *19*, 93–101. [CrossRef]

9. Marichelvam, M.K.; Geetha, M. Application of novel harmony search algorithm for solving hybrid flow shop scheduling problems to minimise makespan. *Int. J. Ind. Syst. Eng.* **2016**, *23*, 467–481. [CrossRef]
10. Deal, D.E.; Yang, T.; Hallquist, S. Job scheduling in petrochemical production: Two-stage processing with finite intermediate storage. *Comput. Chem. Eng.* **1994**, *18*, 333–344. [CrossRef]
11. Jin, Z.H.; Ohno, K.; Ito, T.; Elmaghraby, S.E. Scheduling hybrid flowshops in printed circuit board assembly lines. *Prod. Oper. Manag.* **2002**, *11*, 216–230. [CrossRef]
12. Arthanari, T.S.; Ramamurthy, K.G. An extension of two machines sequencing problem. *Oper. Res.* **1971**, *8*, 10–22.
13. Gupta, J.N. Two-stage, hybrid flowshop scheduling problem. *J. Oper. Res. Soc.* **1988**, *39*, 359–364. [CrossRef]
14. Hoogeveen, J.A.; Lenstra, J.K.; Veltman, B. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *Eur. J. Oper. Res.* **1996**, *89*, 172–175. [CrossRef]
15. Ruiz, R.; Vázquez-Rodríguez, J.A. The hybrid flow shop scheduling problem. *Eur. J. Oper. Res.* **2010**, *205*, 1–18. [CrossRef]
16. Tosun, Ö.; Marichelvam, M.K.; Tosun, N. A literature review on hybrid flow shop scheduling. *Int. J. Adv. Oper. Manag.* **2020**, *12*, 156–194. [CrossRef]
17. Mouzon, G.; Yildirim, M.B. A framework to minimise total energy consumption and total tardiness on a single machine. *Int. J. Sustain. Eng.* **2008**, *1*, 105–116. [CrossRef]
18. Fang, K.; Uhan, N.; Zhao, F.; Sutherland, J.W. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J. Manuf. Syst.* **2011**, *30*, 234–240. [CrossRef]
19. Dai, M.; Tang, D.; Giret, A.; Salido, M.A.; Li, W.D. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robot. Comput. Integ. Manuf.* **2013**, *29*, 418–429. [CrossRef]
20. Luo, H.; Du, B.; Huang, G.Q.; Chen, H.; Li, X. Hybrid flow shop scheduling considering machine electricity consumption cost. *Int. J. Prod. Econ.* **2013**, *146*, 423–439. [CrossRef]
21. Liu, C.H.; Huang, D.H. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *Int. J. Prod. Res.* **2014**, *52*, 337–352. [CrossRef]
22. Lin, W.; Yu, D.Y.; Zhang, C.; Liu, X.; Zhang, S.; Tian, Y.; Xie, Z. A multi-objective teaching−learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint. *J. Clean. Prod.* **2015**, *101*, 337–347. [CrossRef]
23. Ding, J.Y.; Song, S.; Wu, C. Carbon-efficient scheduling of flow shops by multi-objective optimization. *Eur. J. Oper. Res.* **2016**, *248*, 758–771. [CrossRef]
24. Yan, J.; Li, L.; Zhao, F.; Zhang, F.; Zhao, Q. A multi-level optimization approach for energy-efficient flexible flow shop scheduling. *J. Clean. Prod.* **2016**, *137*, 1543–1552. [CrossRef]
25. Lei, D.; Gao, L.; Zheng, Y. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop. *IEEE Trans. Eng. Manag.* **2017**, *65*, 330–340. [CrossRef]
26. Liu, Q.; Zhan, M.; Chekem, F.O.; Shao, X.; Ying, B.; Sutherland, J.W. A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint. *J. Clean. Prod.* **2017**, *168*, 668–678. [CrossRef]
27. Zhang, Y.; Liu, Q.; Zhou, Y.; Ying, B. Integrated optimization of cutting parameters and scheduling for reducing carbon emissions. *J. Clean. Prod.* **2017**, *149*, 886–895. [CrossRef]
28. Lu, C.; Gao, L.; Li, X.; Zheng, J.; Gong, W. A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption. *J. Clean. Prod.* **2018**, *196*, 773–787. [CrossRef]
29. Nasiri, M.M.; Abdollahi, M.; Rahbari, A.; Salmanzadeh, N.; Salesi, S. Minimizing the energy consumption and the total weighted tardiness for the flexible flowshop using NSGA-II and NRGA. *J. Ind. Syst. Eng.* **2018**, *11*, 150–162.
30. Pan, Z.; Lei, D.; Zhang, Q. A new imperialist competitive algorithm for multiobjective low carbon parallel machines scheduling. *Math. Probl. Eng.* **2018**, *2018*, 5914360. [CrossRef]
31. Piroozfard, H.; Wong, K.Y.; Wong, W.P. Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Res. Conser. Recycl.* **2018**, *128*, 267–283. [CrossRef]
32. Wu, X.; Shen, X.; Cui, Q. Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy. *Sustainability* **2018**, *10*, 841. [CrossRef]
33. Meng, L.; Zhang, C.; Shao, X.; Ren, Y.; Ren, C. Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. *Int. J. Prod. Res.* **2019**, *57*, 1119–1145. [CrossRef]
34. Zhou, B.; Liu, W. Energy-efficient multi-objective scheduling algorithm for hybrid flow shop with fuzzy processing time. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2019**, *233*, 1282–1297. [CrossRef]
35. Chen, T.L.; Cheng, C.Y.; Chou, Y.H. Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming. *Ann. Oper. Res.* **2020**, *290*, 813–836. [CrossRef]
36. Wang, S.; Wang, X.; Chu, F.; Yu, J. An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. *Int. J. Prod. Res.* **2020**, *58*, 2283–2314. [CrossRef]
37. Cai, J.; Lei, D. A cooperated shuffled frog-leaping algorithm for distributed energy-efficient hybrid flow shop scheduling with fuzzy processing time. *Complex Intell. Syst.* **2021**, *7*, 2235–2253. [CrossRef]
38. Shi, L.; Guo, G.; Song, X. Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment. *Int. J. Prod. Res.* **2021**, *59*, 576–597. [CrossRef]
39. Wang, J.J.; Wang, L. A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-Shop scheduling. *IEEE Trans. Evol. Comput.* **2022**, *26*, 461–475. [CrossRef]

40. Zuo, Y.; Fan, Z.; Zou, T.; Wang, P. A Novel Multi-Population Artificial Bee Colony Algorithm for Energy-Efficient Hybrid Flow Shop Scheduling Problem. *Symmetry* **2021**, *13*, 2421. [CrossRef]
41. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]
42. Duan, H.; Wang, X. Echo state networks with orthogonal pigeon-inspired optimization for image restoration. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 2413–2425. [CrossRef]
43. Varun, A.; Kumar, M.S. A comprehensive review of the pigeon-inspired optimization algorithm. *Int. J. Eng. Technol.* **2018**, *7*, 758–761.
44. Zhong, Y.; Wang, L.; Lin, M.; Zhang, H. Discrete pigeon-inspired optimization algorithm with Metropolis acceptance criterion for large-scale traveling salesman problem. *Swarm Evolut. Comput.* **2019**, *48*, 134–144. [CrossRef]
45. Goel, S. Pigeon optimization algorithm: A novel approach for solving optimization problems. In Proceedings of the 2014 International Conference on Data Mining and Intelligent Computing, Delhi, India, 5–6 September 2014.
46. Hao, R.; Luo, D.; Duan, H. Multiple UAVs mission assignment based on modified pigeon-inspired optimization algorithm. In Proceedings of the 2014 IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014.
47. Sun, H.; Duan, H. PID controller design based on prey-predator pigeon-inspired optimization algorithm. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014.
48. Zhang, B.; Duan, H. Predator-Prey Pigeon-Inspired Optimization for UAV Three-Dimensional Path Planning. In *Advances in Swarm Intelligence*, 1st ed.; Tan, Y., Shi, Y., Coello, C.A.C., Eds.; Springer: Cham, Switzerland, 2014; Volume 8795, pp. 96–105.
49. Deng, Y.; Zhu, W.; Duan, H. Hybrid membrane computing and pigeon-inspired optimization algorithm for brushless direct current motor parameter design. *Sci. China Technol. Sci.* **2016**, *59*, 1435–1441. [CrossRef]
50. Hu, C.; Xia, Y.; Zhang, J. Adaptive operator quantum-behaved pigeon-inspired optimization algorithm with application to UAV path planning. *Algorithms* **2018**, *12*, 3. [CrossRef]
51. Pei, J.; Su, Y.; Zhang, D. Fuzzy energy management strategy for parallel HEV based on pigeon-inspired optimization algorithm. *Sci. China Technol. Sci.* **2017**, *60*, 425–433. [CrossRef]
52. Rehman, M.H.A.; Javaid, N.; Iqbal, M.N.; Abbas, Z.; Awais, M.; Khan, A.J.; Qasim, U. Demand side management using hybrid genetic algorithm and pigeon inspired optimization techniques. In Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications, Kraków, Poland, 16–18 May 2018.
53. Liu, H.; Yan, X.; Wu, Q. An improved pigeon-inspired optimisation algorithm and its application in parameter inversion. *Symmetry* **2019**, *11*, 1291. [CrossRef]
54. Shang, J.; Li, Y.; Sun, Y.; Li, F.; Zhang, Y.; Liu, J.X. MOPIO: A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection. *Symmetry* **2020**, *13*, 49. [CrossRef]
55. Bolaji, A.L.A.; Okwonu, F.Z.; Shola, P.B.; Balogun, B.S.; Adubisi, O.D. A modified binary pigeon-inspired algorithm for solving the multi-dimensional knapsack problem. *J. Intell. Syst.* **2021**, *30*, 90–103. [CrossRef]
56. Pan, J.S.; Tian, A.Q.; Chu, S.C.; Li, J.B. Improved binary pigeon-inspired optimization and its application for feature selection. *Appl. Intell.* **2021**, *51*, 8661–8679. [CrossRef]
57. Zhun, X.; Liyun, X.; Xufeng, L. An Improved Pigeon-inspired Optimization Algorithm for Solving Dynamic Facility Layout Problem with Uncertain Demand. *Procedia CIRP* **2021**, *104*, 1203–1208. [CrossRef]
58. Hussein, R.B.; Algamal, Z. Solving 0–1 knapsack problem by an improved binary Pigeon Inspired Optimization Algorithm. *Math. Stat. Eng. Appl.* **2022**, *71*, 312–324.
59. Ramalingam, R.; Karunanidy, D.; Alshamrani, S.S.; Rashid, M.; Mathumohan, S.; Dumka, A. Oppositional Pigeon-Inspired Optimizer for Solving the Non-Convex Economic Load Dispatch Problem in Power Systems. *Mathematics* **2022**, *10*, 3315. [CrossRef]
60. Fu, X.; Chan, F.T.; Niu, B.; Chung, N.S.; Qu, T. A multi-objective pigeon inspired optimization algorithm for fuzzy production scheduling problem considering mould maintenance. *Sci. China Inform. Sci.* **2019**, *62*, 1–18. [CrossRef]
61. Wu, X.; Shen, X.; Zhao, N.; Wu, S. An improved discrete pigeon-inspired optimisation algorithm for flexible job shop scheduling problem. *Int. J. Bio-Inspired Comput.* **2020**, *16*, 181–194. [CrossRef]
62. Lei, T.; Luo, C.; Sellers, T.; Rahimi, S. A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping. *Intell. Syst. Appl.* **2021**, *12*, 20053. [CrossRef]
63. Ding, G.; Dong, F. An improved pigeon-inspired optimisation for continuous function optimisation problems. *Int. J. Comput. Sci. Math.* **2023**, *17*, 207–219. [CrossRef]
64. Torky, M.; Gad, I.; Hassanien, A.E. Explainable AI Model for Recognizing Financial Crisis Roots Based on Pigeon Optimization and Gradient Boosting Model. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 50. [CrossRef]
65. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [CrossRef]
66. Marichelvam, M.K.; Prabaharan, T.; Yang, X.S. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Trans. Evol. Comput.* **2013**, *18*, 301–305. [CrossRef]
67. Karthikeyan, S.; Asokan, P.; Nickolas, S. A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints. *Int. J. Adv. Manuf. Technol.* **2014**, *72*, 1567–1579. [CrossRef]
68. Fan, B.; Yang, W.; Zhang, Z. Solving the two-stage hybrid flow shop scheduling problem based on mutant firefly algorithm. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 979–990. [CrossRef]

69. Marichelvam, M.K.; Geetha, M. Solving tri-objective multistage hybrid flow shop scheduling problems using a discrete firefly algorithm. *Int. J. Intell. Eng. Inform.* **2014**, *2*, 284–303. [CrossRef]
70. Chakaravarthy, G.V.; Marimuthu, S.; Sait, A.N. Comparison of firefly algorithm and artificial immune system algorithm for lot streaming in m-machine flow shop scheduling. *Int. J. Comput. Intell. Syst.* **2012**, *5*, 1184–1199. [CrossRef]
71. Marichelvam, M.K.; Geetha, M. A hybrid discrete firefly algorithm to solve flow shop scheduling problems to minimise total flow time. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 318–325. [CrossRef]
72. Qamhan, M.A.; Qamhan, A.A.; Al-Harkan, I.M.; Alotaibi, Y.A. Mathematical modeling and discrete firefly algorithm to optimize scheduling problem with release date, sequence-dependent setup time, and periodic maintenance. *Math. Prob. Eng.* **2019**, *2019*, 8028759. [CrossRef]
73. Lo, H.L.; Fong, S.; Zhuang, Y.; Wang, X.; Hanne, T. Applying a chaos-based firefly algorithm to the permutation flow shop scheduling problem. In Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence, Bali, Indonesia, 7–8 December 2015.
74. Kaya, S.; Gümüşçü, A.; Aydilek, İ.B.; Karaçizmeli, İ.H.; Tenekeci, M.E. Solution for flow shop scheduling problems using chaotic hybrid firefly and particle swarm optimization algorithm with improved local search. *Soft Comput.* **2021**, *25*, 7143–7154. [CrossRef]
75. Rashid, M.F.F.A.; Osman, M.A.H. Optimisation of energy efficient hybrid flowshop scheduling problem using firefly algorithm. In Proceedings of the 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics, Penang, Malaysia, 18–19 April 2020.
76. Tilahun, S.L.; Ngnotchouye, J.M.T.; Hamadneh, N.N. Continuous versions of firefly algorithm: A review. *Artif. Intell. Rev.* **2019**, *51*, 445–492. [CrossRef]
77. Kumar, V.; Kumar, D. A systematic review on firefly algorithm: Past, present, and future. *Arch. Comput. Methods Eng.* **2021**, *28*, 3269–3291. [CrossRef]
78. Ghasemi, M.; Kadkhoda Mohammadi, S.; Zare, M.; Mirjalili, S.; Gil, M.; Hemmati, R. A new firefly algorithm with improved global exploration and convergence with application to engineering optimization. *Decis. Analyt. J.* **2022**, *5*, 100125. [CrossRef]
79. Bacanin, N.; Venkatachalam, K.; Bezdan, T.; Zivkovic, M.; Abouhawwash, M. A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset. *Microprocess. Microsyst.* **2023**, *98*, 104778. [CrossRef]
80. Ezzeldin, R.; Zelenakova, M.; Abd-Elhamid, H.F.; Pietrucha-Urbanik, K.; Elabd, S. Hybrid Optimization Algorithms of Firefly with GA and PSO for the Optimal Design of Water Distribution Networks. *Water* **2023**, *15*, 1906. [CrossRef]
81. Sheeba, A.; Uma Maheswari, B. An efficient fault tolerance scheme based enhanced firefly optimization for virtual machine placement in cloud computing. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7610. [CrossRef]
82. Villaruz, J.A.; Gerardo, B.D.; Gamao, A.O.; Medina, R.P. Scouting Firefly Algorithm and its Performance on Global Optimization Problems. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 445–451. [CrossRef]
83. Paternina-Arboleda, C.D.; Montoya-Torres, J.R.; Acero-Dominguez, M.J.; Herrera-Hernandez, M.C. Scheduling jobs on a k-stage flexible flow-shop. *Ann. Oper. Res.* **2008**, *164*, 29–40. [CrossRef]
84. Nawaz, M.; Enscore, E.E., Jr.; Ham, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* **1983**, *11*, 91–95. [CrossRef]
85. Bean, J.C. Genetic algorithms and random keys for sequencing and optimization. *ORSA J. Comput.* **1994**, *6*, 154–160. [CrossRef]
86. Bureau of Energy Efficiency, Ministry of Power, Govt. of India. Available online: https://cea.nic.in/wp-content/uploads/tpe___cc/2022/02/User_Guide__ver_17_2021.pdf (accessed on 21 October 2022).