algorithms

MDPI

Article

Minimum-Energy Scheduling of Flexible Job-Shop Through Optimization and Comprehensive Heuristic

Oludolapo Akanni Olanrewaju [1], Fabio Luiz Peres Krykhtine [2] and Felix Mora-Camino [1,3,*]

[1] Industrial Engineering Department, Durban University of Technology, Steve Bicko Campus, Durban 4001, South Africa; oludolapoo@dut.ac.za
[2] Industrial Engineering Department, Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro 21941-853, Brazil; krykhtine@poli.ufrj.br
[3] Industrial Engineering Department, Universidade Federal Fluminense, Campus de Rio das Ostras, Rio das Ostras 28895-532, Brazil
* Correspondence: felixmora@id.uff.br

Abstract: This study considers a flexible job-shop scheduling problem where energy cost savings are the primary objective and where the classical objective of the minimization of the make-span is replaced by the satisfaction of due times for each job. An original two-level mixed-integer formulation of this optimization problem is proposed, where the processed flows of material and their timing are explicitly considered. Its exact solution is discussed, and, considering its computational complexity, a comprehensive heuristic, balancing energy performance and due time constraint satisfaction, is developed to provide acceptable solutions in polynomial time to the minimum-energy flexible job-shop scheduling problem, even when considering its dynamic environment. The proposed approach is illustrated through a small-scale example.

Keywords: flexible job shop; scheduling; energy savings; make-span; two-level MILP; heuristics

check for updates

Citation: Olanrewaju, O.A.; Krykhtine, F.L.P.; Mora-Camino, F. Minimum-Energy Scheduling of Flexible Job-Shop Through Optimization and Comprehensive Heuristic. *Algorithms* **2024**, *17*, 520. https://doi.org/10.3390/a17110520

Academic Editors: Francesc Pozo, Alexandre Dolgui, David Lemoine, María I. Restrepo and Frank Werner

Received: 13 September 2024
Revised: 21 October 2024
Accepted: 30 October 2024
Published: 12 November 2024

1. Introduction

The improvement in energy efficiency has not played an explicitly relevant role in the operation of many manufacturing systems in the past since minimizing the make-span has often been the priority. Today, with the increasing prices of energy and the new environmental protection regulations, the energy-saving issue in workshops has become an important research field for universities backed by industrial organizations in advanced countries. The relation between industrial management styles, in particular Lean Management (LM) and its variants, and energy savings has been of recent concern [1], while new formulations of the job-shop scheduling problem, including the issue of energy, have been proposed. Further, authors such as those in [2–4] discussed the possibility of noticeably reducing energy consumption in manufacturing spaces with limited capital investment by rearranging the production process for a given demand through adequate machine selection and operation sequences. Also, in [5], it was shown that, by adjusting the power for each operation in a job-shop environment, relevant energy savings can be obtained. In the case of flexible job shops where machines can competitively perform different operations with differentiated energy costs [6], it is expected that energy-saving opportunities can be made more effective through appropriate machine allocation and the sequencing of operations in the job shop.

1.1. Literature Review

Before introducing the minimum-energy flexible job-shop scheduling problem with the due times considered in this study, an overview of previous instances of job-shop scheduling problems is presented here. The flexible job-shop scheduling problem, at first considered as a mere extension of the job-shop scheduling problem, where it is also necessary to assign machines to the operations of different jobs, has turned out to be, by its added complexity and its adequacy with current industrial practice (Industry 4.0), "a job shop scheduling problem of its own" [7]. In recent years, different reviews covering assumptions, models, and solution techniques for the flexible job-shop scheduling problem have been published [7–11]. Most of the formulations adopted have resulted in Mixed-Integer Linear Programming (MILP) optimization problems with sizes and complexity that often limit the use of exact methods. This has led to the proliferation of heuristics and meta-heuristics, notably nature-inspired ones, which proved rather effective [6,12].

In many studies, the exercise carried out when solving the scheduling problem boils down to delivering a mathematical solution without considering the conditions of its implementation. However, some authors have considered the real-life issues of their implementation by integrating them into the formulation of the scheduling problem to have more effective scheduling: the maintenance of machine condition [13], the availability of machines during production [14], and processing time uncertainties [15] are some of the covered topics. The real environment of production systems is dynamic and is subject to unexpected events such as sudden machine breakdowns, power outages, shortages of materials, the unavailability of operators, and unusual activity durations. Dynamic job-shop scheduling techniques were developed to cope with this reality [16,17]. These techniques are based on three different scheduling policies: predictive, reactive, and predictive–reactive [18]. Predictive approaches build a schedule that is maintained during execution, but when disturbances may appear, the maintenance of its feasibility has a time cost (slack times incorporated into the duration of the operations to absorb delays) and an equipment cost (spare machines to face breakdowns). The scheduling is recomputed with purely reactive approaches after each notable disturbance. Predictive–reactive approaches are a mix of the two previous approaches, where a pre-existent schedule is adapted online during its execution. These techniques increasingly involve tools from artificial intelligence (AI) [19].

When considering energy consumption in a job shop, it is composed of the processing energy consumed by machines; of the transfer energy of raw, semi-finished, or finished products within and around the job shop; of the idle energy consumed by machines during the time intervals between consecutive operations; of the set-up energy of machines to enable the processing of operations; and of the common energy consumed for maintaining acceptable operating conditions in the workshop (lighting, air conditioning, and heating). In flexible job shops, the energy-saving opportunities are not limited to those of job shops, i.e., turning off idle machines, slowing down machine speed, and producing during off-peak periods [20]; they also include the allocation of operations to more energy-efficient machines and the choice of energy-efficient means of transport between machines. This newer and more specialized area of the flexible job shop has been the subject of far fewer survey articles [4,21], even though articles on this subject are published regularly. In these studies, the classical concern has been about minimizing the make-span shifts toward the minimization of energy. Some articles consider a multi-criteria optimization problem with energy and make-span as equal objectives [22–26]; others adopt a single criterion, which can be the weighted sum of energy and make-span [27], the total energy with the make-span as a constraint [28], or the total energy and the make-span as a consequence [3]. Some publications have already addressed the problem of minimum-energy dynamic scheduling in flexible job shops using the predictive reactive approach [23,29].

1.2. The Adopted Approach to Tackle a New Instance of a Job-Shop Scheduling Problem

Considering this literature review and the adopted assumptions of the current study in Section 2, the scheduling problem treated in this study characterizes a sub-class of minimum-energy flexible job-shop scheduling problems for which, to our knowledge, until today, no specific study has been published. The specific characteristics of this problem include the following:

- Release and due time constraints are considered for each job instead of the make-span, which is an objective to be minimized. This differentiates the current problem from, for example, [3,22–28].
- Contrary to all the references consulted, the structures of the jobs are not limited to a mere sequence of activities and allow us to consider complex assembly and de-assembly operations.
- The delays and energy consumption resulting from product transfers between machines are considered. The only reference that considers transfer delays is [26]. However, transfer energy is not considered there.
- The adopted mono-criterion objective function. All the references cited in our literature review for flexible job shop scheduling consider the energy issue, except [3,28], which formulate a multi-criteria optimization problem.

The field of scheduling in industrial production workshops is all the more varied as numerous constraints specific to each realistic situation lead to problems for which different and often new resolution methods and associated algorithms must be developed. The philosophy adopted in this study to deal with a new scheduling problem, as is the case here, is the following:

- First, precisely mathematically formulate the optimization problem and analyze the feasibility of its resolution as a MILP problem using exact standard methods.
- Second, analyze the conditions for implementing a scheduling solution in a dynamic environment. This generally leads to considering the use of approximate resolution methods. At this stage, the generation of a heuristic appears interesting for several reasons: it allows us to obtain, at reduced computational cost, a feasible solution, and it supposes the identification and understanding of relatively simple decision-making mechanisms that can produce acceptable solutions. Heuristics that have been developed for scheduling problems with some common characteristics can be a source of inspiration for its design, and the resolution of their blocking points can be a start for the new heuristic. Once the heuristic has been developed, its performance must be compared with those obtained using exact methods and basic scheduling rules such as priority rules.
- Finally, in the case where it looks interesting to go beyond the performance of the solutions provided by the heuristic, in general by considering the use of metaheuristics (which are much more computation intensive in time and memory than heuristics), the generated heuristic method can be useful. It can provide a starting solution for an available metaheuristic. When a new metaheuristic has to be developed, the heuristic can also give directions for the design of new search mechanisms in the construction of more efficient solutions, or it can even be embedded in the metaheuristics.

1.3. The Objective of the Study

The objective of this study is to contribute to energy efficiency in the manufacturing industry, more particularly in high-energy-consuming integrated, flexible production plants, by developing a new approach to generate energy-efficient schedules with acceptable production delays for flexible job-shop scheduling problems. This study considers the main energy consumption sources in a flexible job shop, machine processing, and transfer of materials. Energy consumed by idle machines is not contemplated. Effectively, if idle times are small, the energy consumed by an idle machine may be negligible, while if idle times are large, the corresponding machines will certainly be shut off.

In this study, jobs are composed of a finite set of operations linked together by precedence–succession constraints without cycling, and due times are assigned to them. Flexibility here refers to the ability to assign the processing of certain operations to different machines. One important element of the solution is the concurrent assignment of the machines to the operations of the different jobs. As early as 1993, Brandimarte [30] considered a two-level approach with the decomposition of the flexible job-shop scheduling problem into routing and job-shop sub-problems to minimize the make-span of a given production plan. His work has received over 400 citations, mainly for his flexible job shop scheduling benchmark and its early use of Taboo Search [31] to solve the two sub-problems than for the proposed decomposition.

In this study, this dual view of the flexible job-shop problem is first adopted to achieve minimum energy schedules by considering the machine assignment of all the operations of a job as a decision variable. The non-consideration of the energy consumed by idle machines means that the energy consumption associated with a production program only depends on the allocation of the operations of each job to the different machines of the job shop and not the timing adopted for their execution. This leads to a formulation of a two-level optimization problem where energy costs and job shop dynamics are separated, opening the way to use exact or approximate solution approaches for this class of mathematical programming problem. This will provide nominal solutions to production plans over a finite period. However, the objectives of this study go well beyond this result. The goal is to propose a scheduling scheme that can adapt easily to cope efficiently with the following:

- Planned (scheduled maintenance, for example) or unplanned disturbances (machine or conveyor breakdown between machines, for example);
- The fast generation of energy-efficient schedules to cope with the arrival of new jobs, allowing permanent operation with the efficient energy performance of the job shop.

This goal has led to the development, in a second step of the study, an ad hoc heuristic able to produce promptly after any significant perturbation or event, an updated solution consistent with the current operational state of the job shop for this original scheduling problem.

The rest of this paper is organized as follows: first, the characteristics of the considered job shop and the adopted notations are displayed in section two; section three presents a separable formulation of a nominal minimum-energy flexible job-shop problem is introduced to ease the search for its exact solution; a heuristic based on the earliest processing time with minimum energy is introduced to generate an open-loop solution to the scheduling problem in the fourth section. Then, the extensions of this heuristic to enable it to cope with disruptions and new jobs are discussed, and the application to a small illustrative case is displayed. Finally, in the Conclusion section, additional research directions are pointed out.

2. The Considered Class of Flexible Job Shops and Their Representation

The considered class of production systems is composed of flexible machines that operate in parallel. There, different products are processed through a subset of production stages in which a machine is assigned to perform a specific operation on a given product. The final products are obtained at the end of the sequences of operations.

2.1. Basic Assumptions

The basic assumptions characterizing the considered flexible job-shop scheduling problem are introduced.

2.1.1. The Plant

The plant is composed of a set of machines, some of which may perform the same tasks. Machines have a fixed position in the workshop, and physical connections exist between some of these machines to allow the transfer of semi-processed products from one machine to the next. The transfer operations are supposedly independent of each other, and it is assumed that there is no traffic congestion in the plant. Then, for each job, the transfer times and energies are constant parameters whose values depend on the position and characteristics of the assigned machines.

2.1.2. The Production Plan

The production plan is composed of independent jobs that can be identical or different; this production plan is known in advance. Each job consists of a set of operations performed on different machines. A directed acyclical graph can represent the precedence/succession constraints between these operations. The introduction of this graph allows one to handle parallel operations within the same job. This does not prevent the repeated use of the same machine for different operations of the same job but assumes that the operations are only executed once per job. This last limitation can be easily overcome by renaming an operation each time, which must be repeated a given number of times within a job. The subset of machines that can carry out each job operation in the workshop is known. For each job of the production plan, a release time and a due time are attached. It is supposed that there are global schedules satisfying the resulting time constraints. Contrary to the Just-In-time scheduling problem, tardiness is not allowed here.

2.1.3. The Operations

The common working period starts at time 0, and its maximum time span is noted as T. All considered jobs have a known release time and a preferred due time. Machines are available for operations since time 0; in the first stage, it is assumed that no breakdowns and maintenance operations occur until time T. Non-dummy operations need only one machine to be performed, machines can only execute one operation at a given time, and pre-emption of machines by other operations is not allowed. A machine can be used more than once by a given job but in different operations. The set-up times of the machines are integrated into the processing times, which, with the transportation times between machines, are considered in this study. Some jobs may have a declared due time that induces a time constraint, which, depending on the context, may be considered hard or soft; the other jobs use T as the due time.

2.1.4. The Objectives

The main objective of this study is to propose a method to minimize the total energy needed to perform a given production plan while satisfying due time constraints for each job. The energy of interest is composed of the processing energy of the different operations on their assigned machines and of the internal logistics energy spent to transport materials from one machine to the next. Here, it is assumed implicitly that there is a unique source of energy, electricity, but this hypothesis could easily be revised.

2.2. Adopted Notations and Representation for Work Plans

The main indexes used to identify the different elements of the sets of interest are as follows:

- i: index of jobs, $i \in \{1, \ldots, n\}$ with $n = |J|$, where J is the set of jobs to be processed.
- k, h: index of operations, $k, h \in \{1, \ldots, S_i\}$ with $S_i = |O_i|$, where $O_i = \left\{O_{i1}, \ldots, O_{S_i}\right\}$ is the set of operations of job i.
- l: index of machines, $l \in \{1, \ldots, m\}$ with $m = |M|$, where M is the set of machines.

To each job i, a directed a-cyclical graph (DAG) is attached, which represents the precedence and succession constraints between its operations. The operations of each job are performed to start and finish with dummy operations of zero duration on the same dummy machines. The operations of a job i are ordered by their increasing rank in G_i. Let Γ_{ik}^{-1} and Γ_{ik} be, respectively, the set of predecessor and successor operations of O_{ik}. The depth of the DAG associated with job i has a depth dr_i, where rank 1 corresponds to the dummy starting operation and rank dr_i corresponds to the dummy closing operation. GL_{ir} is the subset of operations of job i at rank r and $O_i = \oplus_{r=1}^{r=dr_i} GL_{ir}$. Let rt_i and dt_i be, respectively, the release time and the due time of job i, i = 1 to n, where these due times are supposed to be inferior or equal to T. M_{ik} is the subset of machines that can perform operation O_{ik}, δ_{ikl} is the processing time of O_{ik} with its lth machine, and $\tau_{ill'}$ is the transfer time between machine l and machine l' for job i. The nominal transfer time is, in general, the transportation time of the product from one machine to the next, but it can also include other side operations (drying delays, set-up times, and inspection delays). Let pe_{ikl} be the energy necessary for machine l, $l \subset M_{ik}$, to perform O_{ik}, and let $te_{ill'}$ be the energy necessary to transfer product i from machine l to machine l' and other necessary inputs to perform O_{ik} on machine l'.

Figure 1 introduces the DAGs associated with a production plan composed of three jobs. Table 1 shows the machines that can perform the operations of these three jobs, and Figure 2 represents the different possible transfers between successive machines according to the different jobs.

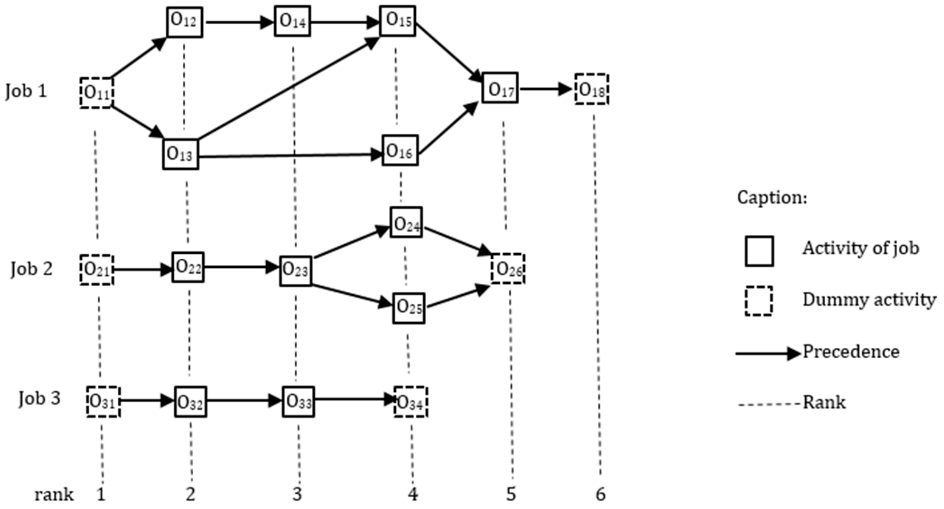

Figure 1. Graphical representation of a production plan.

Table 1. Machines available for operations in production plan.

O_{ik}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{21}	O_{22}	O_{23}	O_{24}	O_{25}	O_{26}	O_{31}	O_{32}	O_{33}	O_{34}
M_{ik}	∅	$\{m_1,m_2\}$	$\{m_1,m_2\}$	$\{m_3,m_4\}$	$\{m_3,m_4\}$	$\{m_5\}$	∅	∅	$\{m_6,m_7\}$	$\{m_2\}$	$\{m_3,m_4,m_5\}$	$\{m_3,m_4,m_5\}$	∅	∅	$\{m_1,m_2\}$	$\{m_6,m_7\}$	∅

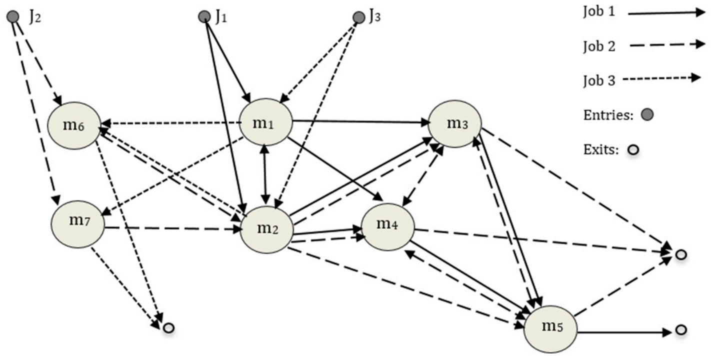

Figure 2. Transfers between machines in a flexible job shop.

The assignment of machines to the successive operations of a job build a unique path between its entry and exit nodes in the directed graph of Figure 2.

The material flow of product under transformation is supposed to be mainly composed of original products to which additional inputs can be aggregated at the level of the operations and from which waste can be retrieved. These aggregations or removals induce costs which, beyond the physical characteristics of the operations, can be related to the position of the machines into the plant. In this study, the different flows of material under processing are taken into consideration. A physical interdependence between the different jobs results from the way the set of machines is shared during the period of operation.

3. Two-Level Formulation of the Minimal-Energy Flexible Job-Shop Scheduling Problem

The flexibility of the job shop presents an additional dimension to the scheduling problem: the assignment of machines to the operations of each job. Technical and functional considerations mean that this choice is limited and multiplies with the number of operations comprising the jobs and with the presence of several machines capable of carrying out the same operations. Here, the assignment of machines to the operations of each job is identified by a decision variable, leading to a two-level formulation of the considered scheduling problem. To accomplish this, additional notations must be introduced.

3.1. Adopted Notations and Representation

Here, A is the set of possible assignments of machines to the operations of all the jobs to be processed, $A = \{A_j,\ j = 1,\ \ldots, |A|\}$, and A_{ij} is the projection of the *j*th assignment on the operations of job *i*. $A_i = \oplus_{j=1}^{|A|} A_{ij}$ is the set of all possible machine assignments for job *i*. The overall knowledge of the allocation of machines to the different jobs is necessary to treat the case of a specific job since these jobs may use subsets of machines concurrently. The index of the machine attached to the processing of O_{ik} in the *j*th machine assignment is written as l_{ikj}.

The maximum number of different assignments of machines for a job *i* is given by

$$n_i^A = |A_i| = \prod_{k=2}^{S_i-1} |M_{ik}| \tag{1}$$

and the maximum number of different machine assignments for the whole production plan is given by

$$n^A = |A| = \prod_{i=1}^{n} n_i^A \tag{2}$$

These numbers can be significantly reduced considering the layout of the plant, the existing transfer system, and the operational state of the machines. In the illustrative case of Section 2.2, we have the following: $n_1^A = 16$, $n_2^A = 18$ and $n_3^A = 4$, leading to n^A = 1152, which is a rather large number. But if adopting the rule that, when two successive operations of the same job can be processed by the same machine, there is no change of machine from the first to the second operation, then the number of accepted machine assignments decreases to 96. In fact, this rule allows one to melt down these pairs of operations into a single one, reducing the size of the scheduling problem.

Let us introduce the processing and transfer times for a given machine assignment: d_{ikj} is the processing duration of O_{ik} by machine l_{ikj}, *i* = 1 to *n*, *k* = 1 to S_i, and *j* = 1 to n^A; and T_{ihkj} is the nominal transfer time of product (*i,h*) from machine l_{ihj} to machine l_{ikj}, *i* = 1 to *n*, *h* = 1 to S_i, $k \in \Gamma_{ih}$, and *j* = 1 to n^A.

When, for machine assignment *j*, a destination machine l_{ihj} of operation O_{ik} is busy with another operation, it is supposed that the semi-processed material of job *i* can wait without using energy until machine l_{ihj} becomes available.

3.2. Absolute Time Bounds

In this subsection, lower bounds for the earliest start times and upper bounds for the latest start times of the operations of each job for a given machine assignment *j* are introduced:

- est_{ikj} is the earliest start time of operation O_{ik} with machine assignment A_{ij} when the release time is respected.
- lst_{kjk} is the latest start time of operation O_{ik} with machine assignment A_{ij} when the due time is respected.

We have the following for job *i* with machine assignment A_{ij}:

$$est_{i1j} = rt_i \text{ and } est_{ikj} = est_{ik-1j} + \max_{h \in \Gamma_{ik}^{-1}} \left\{ est_{ihj} + d_{ihj} + T_{ihkj} \right\}, k = 2 \text{ to } S_i \tag{3}$$

$$lst_{iS_ij} = dt_i - d_{iS_i} \text{ and } lst_{ikj} = \min_{h \in \Gamma_{ik}} \left\{ lst_{ihj} - T_{ikhj} \right\} - d_{ikj}, k = S_i - 1 \text{ to } 1 \tag{4}$$

where rt_i and dt_i are, respectively, the release time and the due time of job *i*.

It can be observed that these earliest and latest start times do not consider the possible concurrent use of the machines by different jobs and the resulting additional delays, so they are, respectively, a lower bound for the earliest start time and an upper bound of the latest start times. Consistency conditions for the time data are

$$est_{ikj} \leq lst_{ikj}, \quad i = 1 \text{ to } n,\ k = 1 \text{ to } S_i,\ j = 1 \text{ to } |A_i| \tag{5}$$

$$lst_{i1j} - d_{i1j} \geq rt_i \quad \text{and} \quad est_{iS_ij} + d_{iS_ij} \leq dt_i \qquad i = 1 \text{ to } n,\ j = 1 \text{ to } |A_i| \tag{6}$$

If a machine assignment *j* for job *i* does not satisfy fully relations (5) and (6), it must be eliminated from A_i.

It is also possible to compute the absolute earliest start time for each operation of each job:

$$Est_{ik} = \min_{j \in A_i}\left\{est_{ikj}\right\} \quad i = 1 \text{ to } n,\ k = 1 \text{ to } S_i \tag{7}$$

It can be observed here that the release time and the due times are not, in general, of the same nature: the release times are independent of the considered process and result from upstream logistics; the due times result from downstream logistics. Then, here, release times are hard constraints, while due times can be a production objective that may not be completely achieved. Thus, the latest starting times and due times constraints can be violated, and the resulting delays characterize a deficit in the production capacity of the job shop to process a given production plan.

3.3. Total Energy Consumed

Here, for a given machine assignment *j*, the energy consumption is considered composed of the energy PE_{ikj} necessary to process operations with machines l_{ikj} and the energy $TE_{ikl_{ihj}j}$ to transport processed products from a machine l_{ikj} to the successor machines l_{ihj} with *h* in Γ_{ik}. Then, the energy used in processing job *i* within machine assignment *j* can be written as

$$E_{ij} = \sum_{k=1}^{S_i} PE_{ikj} + \sum_{k=0}^{S_i} \sum_{h \in \Gamma_{ik}} TE_{ikl_{ihj}j} \qquad i = 1 \text{ to } n,\ j = 1 \text{ to } |A| \tag{8}$$

This expression does not consider the energy consumed by idling machines. Depending on the machine and the durations of its idling periods, it may be interesting to save energy to turn them off when they are not in use rather than leaving them on. Restarting a machine can not only result in a temporary overconsumption of energy but also in a delay of its availability. In this study, it is assumed, since it is generally the case in industrial workshops, that the energy consumption of the idling machines is negligible compared to the processing and transfer energies. So, it does not seem necessary to explicitly introduce this component of the energy spent in a mathematical formulation of the scheduling problem, especially since this would lead to increased complexity resulting from the new variables and constraints to be introduced. What can be achieved—once a solution to the overall scheduling problem has been obtained that minimizes the main components of the consumed energy, whether this solution is exact or approximate—is, in a second step, to evaluate the idle periods of each machine and decide whether to turn them off or keep them on until their next use. This is the kind of problem that could be solved by an expert system using artificial intelligence techniques to exploit both the characteristics of the scheduling solution and the energy performance of each machine.

3.4. The Two-Level Minimum-Energy Scheduling Problem

The minimum-energy flexible job-shop scheduling problem formulated as an optimization problem by using the machine assignment variables x_{ij}, the precedence variables $z_{iji'j'kk'}$ and the timing variables t_{ijk} is the following:

x_{ij} = 1 if machine assignment *j* is chosen to perform job *i*, and x_{ij} = 0 otherwise;

t_{ikj}: the starting time of operation O_{ik} at machine l_{ikj};

$z_{iji'j'kk'}$ = 1 if operation O_{ik} with machine l_{ikj} is decided to be realized before operation $O_{i'k'}$ using the same machine (l_{ikj} = $l_{i'k'j'}$), and $z_{iji'j'kk'}$ = 0 otherwise.

The problem is formulated as a mono-criterion optimization problem, avoiding the introduction of relative weights between criteria or the exploration of a Pareto frontier involving energy and make-span. The global criterion to be minimized is expressed as

$$\min_{x,t,z} E_T = \sum_{i=1}^{n} \sum_{j=1}^{|A_i|} x_{ij} \cdot E_{ij} \tag{9}$$

The considered constraints are

$$\sum_{j \in A_i} x_{ij} = 1 \qquad i = 1 \text{ to } n \tag{10}$$

and for ij , such as $x_{ij} = 1$:

$$est_{i1j} \leq t_{i1j} \;\; i = 1 \text{ to } n \tag{11}$$

$$\max_{h \in \Gamma_{ik}^{-1}} \left\{ t_{ihj} + d_{ihj} + T_{ihj} \right\} \leq t_{ikj} \; i = 1 \text{ to } n, k = 2 \text{ to } S_i \tag{12}$$

$$t_{iS_ij} + d_{iS_ij} \leq dt_i \; i = 1 \text{ to } n \tag{13}$$

For ij and $i'j'$, such as $x_{ij} = x_{i'j'} = 1$:

$$\forall i \neq i',\ \forall j, j',\ \forall k,\ k' \text{ with } l_{ikj} = l_{i'k'j'} \;\;:\;\; z_{iji'j'kk'} + z_{i'j'k'k} = 1 \tag{14}$$

$$\forall i \neq i',\ \forall j, j',\ \forall k,\ k' \text{ with } l_{ikj} \neq l_{i'k'j'} \;\;:\;\; z_{iji'j'kk'} = z_{i'j'k'k} = 0 \tag{15}$$

and $\forall i \neq i'$, $\forall j, j'$, $\forall k$, k' with $l_{ikj} = l_{i'k'j'}$:

$$t_{ikj} + d_{ikj} \leq t_{i'k'j'} + V\left(1 - z_{iji'j'kk'}\right) \text{ and } t_{ikj} + d_{ikj} + V \cdot z_{iji'j'kk'} \geq t_{i'k'j'} \tag{16}$$

The significance of the different constraints are as follows: constraint (10): the choice of a unique path for each job; constraint (11): feasibility bounds for the starting time of the first operation of job i with machine assignment j; constraint (12): starting time succession constraints along a path; constraint (13): due time constraints for the different jobs, where $t_{iS_ij} + d_{iS_ij}$ is the completion time of job i along path j; constraints (14), (15), and (16): non-overlapping of operations of chosen paths on the same machine, where V is a very large number. However, in situations in which the feasible time intervals for processing on a same machine (l_{ikj} = $l_{i'k'j'}$) do not intersect, the value of the z variable is already fixed:

$$lst_{ikj} + d_{ikj} \leq est_{i'k'j'} \;\;:\;\; z_{iji'j'kk'} = 1 \tag{17}$$

$$est_{ikj} \geq lst_{i'k'j'} + d_{i'k'j'} : \;\; z_{iji'j'kk'} = 0 \tag{18}$$

Problem (9)–(16) is a two-level optimization problem, where the machine assignment problem, with the objective of minimizing total energy consumption, is considered at the upper level (relations (9) and (10), with only decision variable x_{ij}), while at the lower level, the satisfaction of dynamic constraints (11)–(16) lead to a feasible scheduling. Different approaches appear of interest to generate solutions for instances of this problem, which is expected to be of the NP-Complete complexity class [32]. The transformation of formulation (9)–(16) into an MILP-type formulation greatly increases the number of variables and constraints, making its exact numerical resolution difficult. In [33], a greedy heuristic was developed to provide, within the adopted formalism, a feasible sub-optimal solution for a simplified version of this problem: there, the DAGs of the jobs reduced to single chains, which were selected repeatedly according to their energy performance until the due dates were satisfied. To design exact solution algorithms, Branch and Bound strategies appear here to be of direct interest. Also, Benders decomposition-based algorithms, developed to solve general two-level optimization problems, could be particularized to this problem [34]. The above problem can also be seen as a simplified case of a bi-level MILP problem. In that last field, many exact [35,36] and approximate [37,38] methods have been proposed in the recent literature. Relevant work could improve the computational efficiency of some of these methods when applied to problems (9)–(16).

3.5. Evaluation of Solution Performance

The solution of the above problem produces the make-span:

$$MS = \max_{\forall i, \forall j \in A_i} \left\{ \left(t_{iS_ij} + d_{iS_ij} \middle| x_{ij} = 1 \right) \right\} - \min_{\forall i, \forall j \in A_i} \left\{ \left(t_{i1j} \middle| x_{ij} = 1 \right) \right\} \tag{19}$$

In Equations (14)–(16), the decision variables x_{ij} are absent; then, by choosing for each t_{ijk} with $x_{ij} = 0$, their earlier starting time, constraints (14)–(16) are satisfied by these t_{ijk} variables. Then, in the case in which all the release time are to zero, (19) can be rewritten more simply:

$$MS = \max_{\forall i, \forall j \in A_i} \left\{ t_{iS_ij} + d_{iS_ij} \right\} \tag{20}$$

In the general case, the make-span for machine l is given by

$$MS_l = \max_{i,k,j} \left\{ t_{ikj} \middle| x_{ij} = 1 \text{ and } l_{ikj} = l \right\} - \min_{i,k,j} \left\{ t_{ikj} \middle| x_{ij} = 1 \text{ and } l_{ikj} = l \right\} \tag{21}$$

and the total processing time of machine l is given by

$$PT_l = \sum_{i=1}^{n} \sum_{j \in A_i} x_{ij} \sum_{k \in \{1,\ldots,S_i\},\ l_{ikj} = l} d_{ikj} \tag{22}$$

Assuming that the machines once started are shut down only when their last operation is completed, their idle time where a residual amount of energy is used is given by

$$IT_l = MS_l - PT_l \tag{23}$$

It can be observed that the resolution of problem (9)–(16) will provide a scheduling of minimum total energy without considering the consequences for its make-span. If the jobs had no due times and the working period [0, T] was sufficiently large, the solution of problem (9)–(16) will consist of selecting each job's machine assignment of minimum energy E_{ij}. A way to introduce the make-span minimization objective can be obtained by setting, at smaller values, the due times of the different jobs or their global maximum value $\max_i \{dt_i\}$ according to the production context. Successive resolutions of the problem with decreasing due times will make it possible to highlight the influence of the make-span on the energy performance of the workshop for a given production plan J. This influence should be analyzed to enable the identification of characteristic patterns within some families of production plans, which will allow, once again by using artificial intelligence techniques such as Machine Learning [39,40], to guess correctly the feasibility of the adopted due times before embarking on the numerical resolution of the optimization problem.

4. Minimum-Energy Scheduling Heuristic with Due Times

The previous section considered a deterministic situation, and the solution of the minimum-energy flexible job-shop will provide the best schedule to follow during nominal conditions. The analysis of optimal solutions makes it possible to evaluate the performance of the job shop and identify its blocking points, eventually leading to its redesign or resizing. Then, different configurations of the job shop can be assessed through the optimal solution of the scheduling problem by considering a set of typical production plans. However, when considering the very common situation in which the workshop is subject to disturbances (delays, malfunctioning or breakdowns of machines, and unavailability of operators), to postponement of jobs, or to the arrival of new jobs to be processed in the same period, the optimal schedule is no more completely feasible. Until a new stabilized situation is established in the job shop and a new optimal schedule is established, short-term local decisions should be taken to keep the job shop in operation. In the case in which perturbations occur frequently, it will be necessary to establish a much more responsive system. Also, when the job shop is working full time, the segmentation of production into finite production plans should lead to sub-optimal schedules. Considering the limitations of this approach, an open-loop heuristic to cope, on a reactive basis with the minimum-energy flexible job-shop scheduling problem, is introduced in this section. In the first stage, the heuristic is developed to cope with a given production plan where jobs have release and due times; then, it is shown how to extend its application to the dynamic environment of workshops.

4.1. Adopted Principles to Design the Heuristic

Here, the main idea is to develop, on a moving time front line, the allocation of machines and time schedules at the earliest start time of operations in ascending rank in any of the active jobs. This greedy scheme will promptly produce a solution promoting energy savings that is feasible for the current situation of the job shop. This will allow a reactive approach to reschedule the current production plan as many times as necessary. The solution must face the dilemma of having to choose between short job processing times and minimizing the necessary energy. This is achieved as follows: on the one side, the earliest assignment of a machine to a new operation is performed, but on the other side, the choice between the available machines is limited to the more energy efficient ones. The decision to assign a machine and to schedule the starting time for an operation will be based, on one side, on the processing delays and transfer delays from its directly preceding operations and, on the other side, on the transfer energy from the preceding operations and on the processing energy of the candidate machine. The driving dimension in the heuristic is time, since this will yield feasible solutions constructed step by step in the timeline according to the earliest feasible starting time for the operations. To assess, according to energy, the transitions from one operation to a successor one, new parameters must be defined. For operations O_{ih} and O_{ik} such that $O_{ik} \in \Gamma_{ih}$, the energy cost of deciding that O_{ik} will be processed by machine l', while O_{ih} is performed by machine l_h, is the sum of the transfer energy and of the process energy $tpe_{ihkll'}$ given by $tpe_{ihkll'}$=$\sum_{h\in\Gamma_{ik}^{-1}} te_{il_h l'} + pe_{ikl'}$ (see Figure 3).

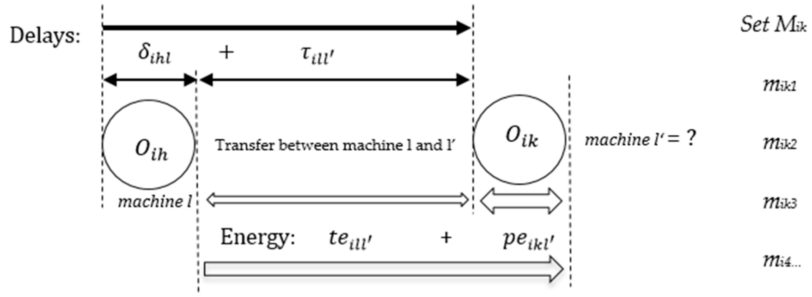

Figure 3. Delays and energy costs for machine assignment to O_{ik}.

Consider now the sets of machines $MM_{ih} = M_{ih} \times \cup_{k\in\Gamma_{ih}^{-1}} M_{ik}$, h = 1 to S_i and their associated energy costs. In large job shops, several machines may be identical, with almost equal transport time and energy between some pairs of machines. Pairs (l, l') with the same total transfer energy costs can be grouped into the same subset, and these subsets can be ranked by increasing energy costs. Let MM_{ih}^{w} be these sets, with w_i = 1 to W_i, where $MM_{ih} = \oplus_{w_i=1}^{w_i=W_i} MM_{ih}^{w}$. Here, MM_{ih}^{1} is the subset of pairs with the minimum total (transportation and processing) energy costs, while $MM_{ih}^{W_i}$ is the subset of pairs with the highest energy cost.

It is important to note that the operations that can be candidates to be assigned a machine are those for which all their predecessor operations have already been assigned and scheduled. Then, the operations of each job are addressed by increasing rank in the corresponding DAG. The scheduling approach is a greedy one, so idle periods of machines may happen, but on the other side, decisions about machines will only consider their next free time up to the end of the working period, which is much easier to treat. It is important to also observe that, contrary to the approach presented in the previous section, in this case, the machine assignment to operations is performed step by step in the timeline.

4.2. Algorithm of the Scheduling Heuristic

The algorithm is composed of three stages: The first stage initializes the values of the different variables and sets, and the second stage assigns a machine and a time schedule to the earliest possible operation of any job. This stage is repeated until all the operations of all the jobs have been assigned a machine and scheduled a processing time. In the third stage, the feasibility of the obtained solution is assessed to conclude the assignment or to widen the set of machines to perform critical operations.

Adopted notations for varying quantities and sets used by the algorithm in the solution-building process include the following:

Γ_{ik}^{U}: the set of predecessors of operation k in job i, which have not been used to process O_{ik}. Γ_{ih}^{D} : the set of successor operations of operation O_{ih} in job i, which have not been processed. GL_{ir} is the subset of operations of job i at rank r. GU_{iu}: the set of operations of job i at rank u in $\mathrm{DAG}_{\mathrm{i}}$, which have unprocessed successor operations. GD_{iu}: the set of operations of job i at rank u in $\mathrm{DAG}_{\mathrm{i}}$, which are not completely processed. $nft(l)$: the next free time of machine l until the end of the working period. J_a: the set of active jobs, i.e., jobs whose activities have not been completely processed.

Figure 4 shows a flowchart of the proposed heuristic, which is described in detail afterward in Algorithm 1.

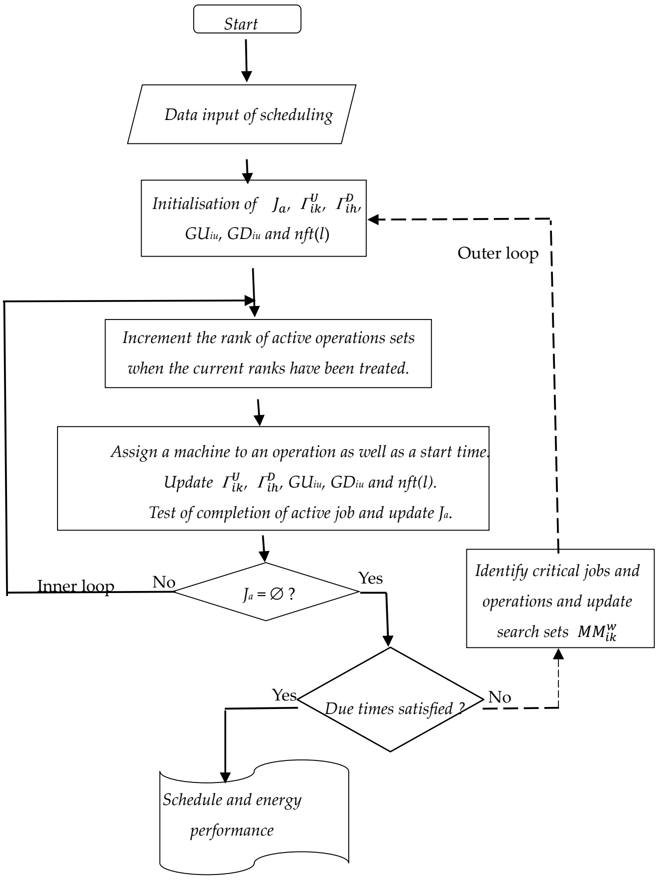
Start
Data input of scheduling
Initialisation of J_a, Γ_{ik}^{U}, Γ_{ih}^{D}, GU_{iu}, GD_{iu} and $nft(l)$
Outer loop
Increment the rank of active operations sets when the current ranks have been treated.
Assign a machine to an operation as well as a start time. Update Γ_{ik}^{U}, Γ_{ih}^{D}, GU_{iu}, GD_{iu} and $nft(l)$. Test of completion of active job and update J_a.
Inner loop
No
$J_a = \varnothing$?
Yes
Identify critical jobs and operations and update search sets MM_{ik}^{w}
Yes
Due times satisfied ?
No
Schedule and energy performance

Figure 4. Flowchart of proposed heuristic.

Algorithm 1: Proposed Scheduling Algorithm

```
(0) Initialization of the solution process.
  $i = 1$
  While $i \leq n$ do
              $k = 1,\ t_{ik} = rt_i,\ \delta_{ik} = 0, w_i = 1, u_i = 1, u = 1, l = 1;$
              While $u \leq dr_i$
                  $GU_{iu} = GL_{iu}$ and $GD_{iu} = GU_{iu}$
              End While $u$
                  While $k \leq S_i$
                      $t_{ik} = Est_{ik}$ and $\Gamma_{ik}^{D} = \Gamma_{ik}$ and $\Gamma_{ik}^{U} = \Gamma_{ik}^{D}$
                  End While $k$
  End While $i$
  While $l \leq m$ do
              $nft(l) = 0$
  End While $l$
    $J_a = \{1,\ 2,\ \ldots, n\}$ is the set of active jobs.
End of initialization
(1) Inner loop: Iteration on the schedule of the next operation and selection of machine to
    process it.

  While $J_a \neq \varnothing$ do
          Shift of rank in the DAGs:
              $\forall i \in J_a$ : if $GD_{iu_i} = \varnothing$, increment $u_i$: $u_i = u_i + 1$.
        Select the next job, operation to be processed and machine on the time line:
          Choose $(i^*,\ h^*, k^*, l^*)$ such as:

    $(i^*,\ h^*, k^*, l^*) = \underset{i \in J_a,\ O_{ih} \in GU_{iv},\ 1 \leq v < u_i, O_{ik} \in\ GD_{iu_i}}{\operatorname{argmin}} \left\{ \max_{(l_{ih}, l) \in MM_{ik}^{w_i}} \{ t_{ih} + \delta_{ihl_{ih}} + \tau_{il_{ih}l},\ nfl(l) \} \right\}$   (24)

        Update the subsets of active operations:
        $\Gamma_{i^*h^*}^{D} = \Gamma_{i^*h^*}^{D} - \{O_{i^*k^*}\}$ and if $\Gamma_{i^*h^*}^{D} = \varnothing :\ GU_{iu_{i^*-1}} = GU_{iu_{i^*-1}} - \{O_{i^*h^*}\}$
  $\Gamma_{i^*k^*}^{U} = \Gamma_{i^*k^*}^{U} - \{O_{i^*h^*}\}$ and if $\Gamma_{i^*k^*}^{U} = \varnothing :\ GD_{iu_{i^*}} = GD_{iu_{i^*}} - \{O_{i^*k^*}\}$
        Machine assignment and start time update:
The machine assigned to $O_{i^*k^*}$ is $l_{i^*k^*} = l^*$ and its earliest start time is updated to:

        $t_{i^*k^*} = \max\{ t_{i^*k^*},\ max\{ t_{i^*h^*} + \delta_{i^*h^*l_{i^*h^*}} + \tau_{i^*l_{i^*h^*}l},\ nfl(l^*)\ \}\ \}$   (25)

          Update next free time of machine $l_{i^*k^*}$:

              $nft(l_{i^*k^*}) = t_{i^*k^*} + \delta_{i^*k^*l_{i^*k^*}}$   (26)

          Closing job $i$:
              If $k_{i^*} = S_{i^*}$ and $\Gamma_{i^*k^*}^{U} = \varnothing$, job $i$ is completed, then: $J_a = J_a - \{i^*\}$
    End While $J_a$
```

For a given production plan, the finite number of iterations is upper-bounded by the sum of the number of operations in the different DAGs, and the finite number of operations inside an iteration is majored by $n \cdot S^2 \cdot \mu$, where $S = \max_{i=1\ to\ n} S_i$ and $\mu = \max_{i,k} \left| MM_{ik}^{w_i} \right|$ which is upper-bounded by m^2. Then, the time complexity of this heuristic is polynomial (P). This heuristic can therefore claim to provide, within a reduced computational time, a feasible solution, aimed at energy savings, to production plans with large dimensions (the number of jobs, the number of operations per job, and the number of machines).

4.3. Illustration of the Assignment and Scheduling Process of the Proposed Heuristic

To illustrate the core of the new assignment process represented by relation (24), Figure 5 represents a situation after seven machine assignments and scheduling of the operations of the jobs considered in Section 2.2. For the eighth iteration of the inner loop of the algorithm, the active rank of job 1 is 4, with candidate activity O_{15} for machine assignment and scheduling; for job 2, the active rank is also 4, with candidate activities O_{24} and O_{25}; and finally, for job 3, the active rank is 3, with candidate activity O_{33}.

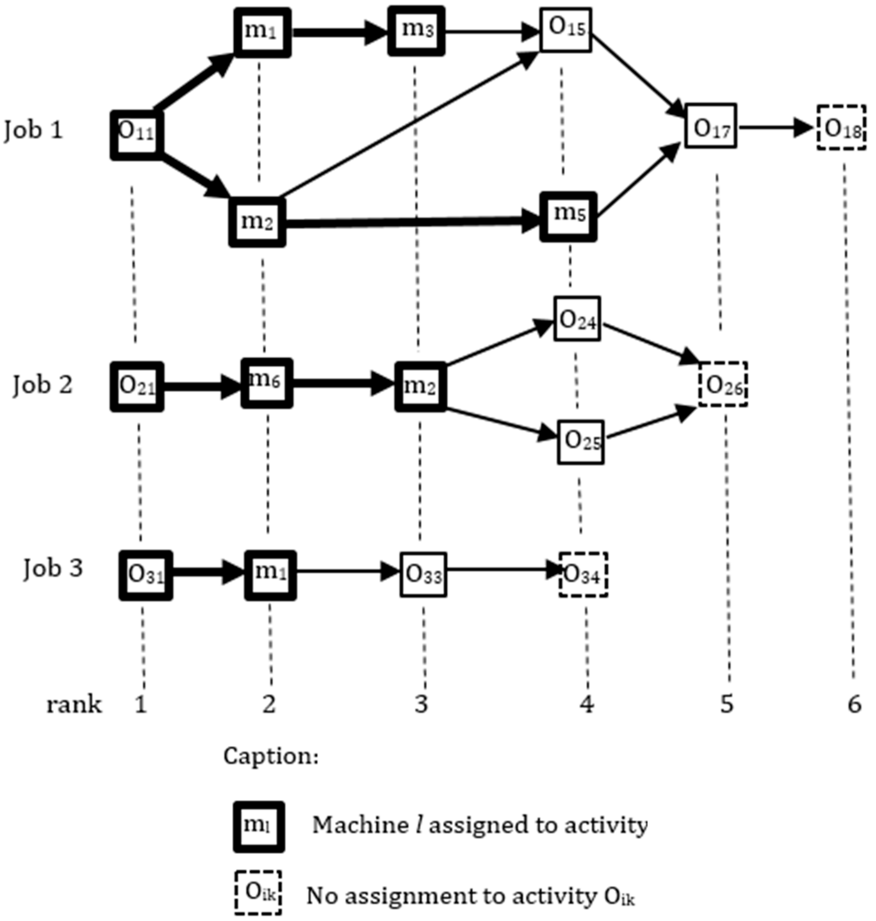

Figure 5. Current machine assignment after 7 iterations of the inner loop.

Here, only numerical values necessary to understand the selection process of Equation (24) are introduced. Table 2 gives the next free times for the seven machines; Table 3 gives the end time of already scheduled and assigned operations, and Table 4 gives the added delays and energies resulting from a machine assignment to each candidate operation in the active rank of each job.

Table 2. Current next free times for the machines in the job shop.

Machine *l*	1	2	3	4	5	6	7
nft (*l*)	0+12	0+14	0+17	0	0	0+7	0

Table 3. End time of assigned parent operations.

Operations	O_{15}	O_{24}	O_{25}	O_{33}
O_{13}	17	------	------	------
O_{14}	5	------	------	------
O_{23}	------	12	12	------
O_{32}	------	------	------	7

Table 4. Added delays/energies with machine assignment.

	O_{15}	O_{24}	O_{25}	O_{33}
m_3	8/100	12/80	10/120	------
m_4	10/80	8/140	10/100	------
m_5	------	10/100	6/140	------
m_6	------	------	------	14/60
m_7	------	------	------	12/90

Now, assuming that the sets MM_{ik} for the candidate operations are subdivided in two subsets according to the spent energy and are given by $MM_{15}^{1} = \{m_4\}$, $MM_{15}^{2} = \{m_3\}$, $MM_{24}^{1} = \{m_3, m_5\}$, $MM_{24}^{2} = \{m_4\}$, $MM_{33}^{1} = \{m_6\}$, and $MM_{33}^{2} = \{m_7\}$, the earliest start time for O_{15} is *max*{0, 17 + 10}; for O_{24}, it is *min*{*max*{14, 12 + 8}, *max*{14, 12 + 10}}; and for O_{33}, it is *max*{0, 7 + 14}. Then, at this stage, relation (24) produces i^* = 3, k^* = 3, h^* = 2, and l^* = 3, i.e., at this stage, operation O_{33} is scheduled at t_{33} = 20, and machine m_3 is assigned to process it.

The local decision rule of this heuristic, which first addresses the temporal constraints by considering the most quickly available machines to perform an operation, before making a choice based on energy consumption, seems adapted to the situation of flexible workshops where different machines can perform the same operations.

4.4. The Resulting Solution, Assessment, and Adaptation (Outer Loop of Algorithm)

The solution (written without the asterisk symbol) provides, for each job, the starting time and the assigned machine for each operation: t_{ik} and l_{ik} for $k = 2$ to $S_i - 1,\ i = 1$ to n.

The completion time of job i is $ct_i = t_{iS_i} + \delta_{iS_i l_{S_i}}$ and the energy necessary to perform the jobs is given by

$$E = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} \sum_{k=2}^{S_i-1} ep_{ikl_{ik}} + \sum_{h=1}^{S_i-1} \sum_{k \in \Gamma_{ih}} te_{il_{ih}l'_{ik}} \tag{27}$$

If the following conditions are satisfied,

$$ct_i \leq dt_i \quad i = 1 \text{ to } n \tag{28}$$

the proposed solution composed of the selected machines and schedules is feasible.

When one or more conditions (28) are not satisfied, it means that some jobs are over-delayed in queues at machines corresponding to minimum-energy steps, and a new feasible solution must be found. For that, let us widen the search process to less energy-efficient machines to process critical jobs in the current solution. These critical jobs include

$$I_c = \left\{ i \in I \text{ with } ct_{ij} > dt_i \right\} \tag{29}$$

Considering the succession of restrictions in the process of a job and the competition for energy efficient machines, the difference t_{ik}-Est_{ik} is expected to increase with k for a given job i. However, a large variation of this difference from an operation to the next can be interpreted as the presence of a queue to use machine l_{ik}. Then, for the operation

O_{ik}, the search for an energy-efficient machine will be relaxed by increasing the index w_i and searching in sets $MM_{ik}^{w_i}$ of relation (24). To avoid a large degradation in the energy performance, this will be performed for the most critical operation k_{ic} of each critical job, where k_{ic} is given by

$$k_{ic} = \underset{k, 2 \leq k \leq S_i - 1}{\text{argmax}} \{(t_{ik+1} - Est_{ik+1}) - (t_{ik} - Est_{ik})\} \qquad i \in I_c \tag{30}$$

and the search for the next free machine to perform operation $O_{ik_{ic}}$ will be performed now over the sets

$$MM_{ik_{ic}}^{w_i'} \text{ with } w_i' = w_i + 1 \quad i \in I_c \tag{31}$$

Another solution could be, when the machines have different regimes of operation, to choose faster regimes of operation in the critical sets $MM_{ik_{ic}}^{w_i}$. The search process must be restarted until conditions (28) are met. If, after many iterations of this process, conditions (28) are not met and no progress is achieved with respect to the completion times, the current solution, which is a feasible one with respect to all the other constraints, should be adopted, and the current completion times should be considered as new due times.

4.5. Dynamic Scheduling with Heuristic

What can be accomplished to face a dynamic workshop environment with the objective of preserving the energy efficiency of the computed schedule depends on the nature and the magnitude of the disruption:

4.5.1. Small Delays

To cope with small delays that are observed or expected in the processing of operations, once a scheduling solution is obtained, it should be of interest to compute from it the current time margins (total, free, and certain margins) and to check if the delay can be absorbed by the margin of interest. If this is not sufficient, some machines operating, or operating a late operation, may be speed up, but in general, at an increase of energy cost.

4.5.2. Programmed Unavailability of Machines

To cope with programmed unavailability of machines (maintenance of condition, for example), the production plan can include a dummy job whose unique activity is the maintenance of the machine with release and due times corresponding to the planned maintenance period.

4.5.3. Sudden Breakdown of Machines

To cope with a sudden breakdown of a machine that is or is not in operation, first estimates of its reparability and, if repairable, its repair time, must be performed. Then, if the resulting delay until repair cannot be absorbed by time margins and, if other machines able to perform the directly impacted operations are not available, a complete rescheduling must be computed with the new expected situation during the work period.

4.5.4. Modification of the Production Plan

To cope with a new arriving order with jobs during the current working period, one option could be to use the idle slots of the machines in the current schedule, but with the risk of choosing the less energy efficient ones that have been bypassed by the current optimal schedule. Another option, often the only feasible one, is to compute a new solution for problem (9)–(16) where the current state of the jobs is updated to their current state of completion: the remaining operations of jobs partially processed, the release times of operations currently under processing, and the next free time for a machine under operation or under maintenance or repair. The rationale of the proposed heuristic is that the scheduling up to the release date of the new jobs remains unchanged, avoiding disruptions in the job shop operations. In the case of a continuum of new job arrivals, the rationale of the proposed heuristic, the earliest start time, should limit the accumulation of late jobs. The responsiveness of this last solution will depend not only on the calculation time of the new schedule but mostly on the time necessary to update the input data of the problem.

5. Evaluation of the Proposed Scheduling Heuristic

The purpose of this section is to comprehensively assess the behavior of the proposed heuristic by comparing it with priority rules for flexible job shop problems. The operation of the proposed heuristic was described in detail in Section 4.3 through an example. Several benchmarks are available in the literature [41] to compare heuristics for flexible job-shop problems, particularly minimizing the make-span of production plans. However, in these benchmarks, due times for jobs, as well as costs and delays associated with transfers between machines, are not considered in general. Since the operation of the proposed heuristic is totally deterministic, it seemed to us that instead of carrying out a cumbersome statistical evaluation, as should be performed with complex metaheuristic involving random processes, it seemed more judicious to us to explain why this heuristic produces better results than priority rules of the same class, i.e., greedy deterministic constructive ones, where the schedule solution is constructed step by step from scratch through deterministic processes according to a local evaluation of a criterion [42].

5.1. Priority Rules for Flexible Job Shop Scheduling

Simple priority rules were developed first for jobs processed by a single machine and then for jobs composed of a string of operations to be processed on given machines in a job shop; a survey classifying tens of them is given in [43]. When the objective is to minimize the make-span of a production plan, one of the more intuitive priority rules is the Shortest Processing Time (SPT) algorithm, which prioritizes operations based on their processing times, with the shortest operations being handled first. Since the focus in this study is on minimizing total energy, a new basic rule, a counterpart of SPT, can be introduced: the Smaller Processing Energy (SPE), whose algorithm prioritizes tasks based on their processing energy, with the more energy-efficient tasks being selected first. However, when considering flexible job shop scheduling, this is not sufficient, since some operations can be processed concurrently on different machines. Then, in this case, a priority rule must address two sub problems [44]: machine assignment and job sequencing. In the schedule of a flexible job shop, when an operation of a job is planned to finish to be processed on a machine at some scheduled time, the schedule must provide two pieces of information:

- Which machine will process the next operation of that job?
- What will be the next operation of a job processed by that machine?

The successive answers to the first question can describe the processing of the jobs according to the schedule. In contrast, the successive answers to the second question can describe the load plan of each machine according to this same schedule. The answer to the second question results from the choice of an operation among the ones waiting at that machine. If the queue of operations at that time is empty, the machine becomes idle until new operations are assigned to it. Simple priority rules will tackle these two decision problems in sequence and not simultaneously, contrarily with what is achieved in the heuristic proposed in this study with relation to (24). Then, in the case of a flexible job shop, a priority rule will have first to assign, for each operation, a machine, and then schedule each operation on its assigned machine. It is important to observe that assigning each operation a priori to a machine according to its processing time or processing energy for this operation would result in priority rules that are much simpler to implement, but this could generate additional delays since it would be necessary to wait for the machine previously assigned to an operation to become free, while other machines may be available earlier. Then, two on-line priority rules, which can be seen as extensions of SPT and SPE to the flexible case of job shop scheduling, are considered for comparison with the proposed heuristic:

- One focusing on processing time, named TTE (indicating Time–Time–Energy according to the nature of the criterion considered at each stage of the priority rule), where the machine is able to process earlier that operation is assigned to it and where the operation to be next processed among the waiting operations of a machine is chosen according to SPT. When different machines have the same fastest processing time for an operation, the one with the smallest processing energy will be assigned to it, and in case of equal energy performance, a random choice will be made among them. If different queuing operations at the same machine have the same processing time, the FIFO rule is applied.
- One focusing on processing energy, named ETT (for Energy–Time–Time), where a machine is able to process with the minimum energy that operation is assigned to it, while the operation to be next processed among the waiting operations of a machine is also chosen according to SPT. In the case of machines with equal processing energy performance, the one with the shortest processing time will be assigned to it, and in the case of an equal processing time, a random choice will be made among these waiting operations. If different queuing operations at the same machine have the same processing time, the FIFO rule is also applied.

5.2. Comparison of Performances Between the Heuristic and the Priority Rules

Figure 6 displays a local situation where several operations, represented by black dots, are candidates to be assigned to different machines according to the proposed heuristic (called HET), the ETT, and the TTE heuristics. There, the abscissa and the ordinate inform, respectively, for each candidate solution, its additional energy cost (δe) and resulting delay (δd). According to the description of the three different decision processes, the heuristic selects a local solution that has intermediate performance in terms of additional energy and delay ($\delta e_{ETT} \leq \delta e_H \leq \delta e_{TTE}$ and $\delta d_{TTE} \leq \delta d_H \leq \delta d_{ETT}$).

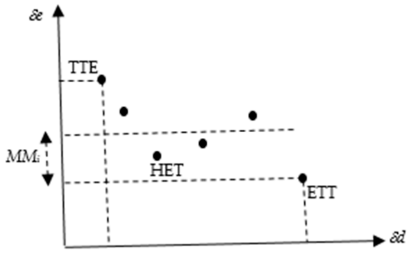

Figure 6. Example of local decision space.

To illustrate the propagation of this property, let us consider a small flexible job shop, displayed in Figure 7, with three machines (m_1, m_2, and m_3) with two jobs (J_1 and J_2), composed of two operations (O_{11} and O_{12}) and (O_{21} and O_{22}). The operations O_{11} and O_{21} can be performed either with machine m_1 or with machine m_2, while operations O_{12} and O_{22} are performed with machine m_3. The processing and transfer costs (delays and energies) are given by d/e in the figure.

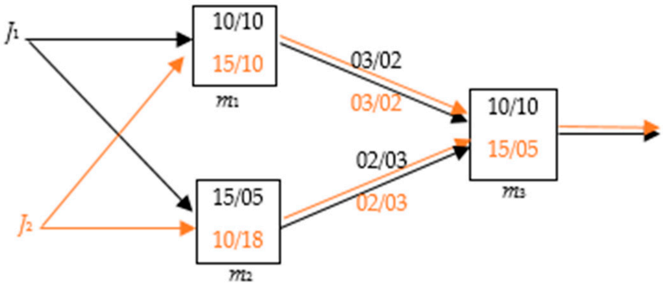

Figure 7. Small flexible job shop (job 1 in black; job 2 in orange).

The application of the heuristic and the two priority rules leads to the schedules displayed in Figure 8, with the performances displayed in Table 5.

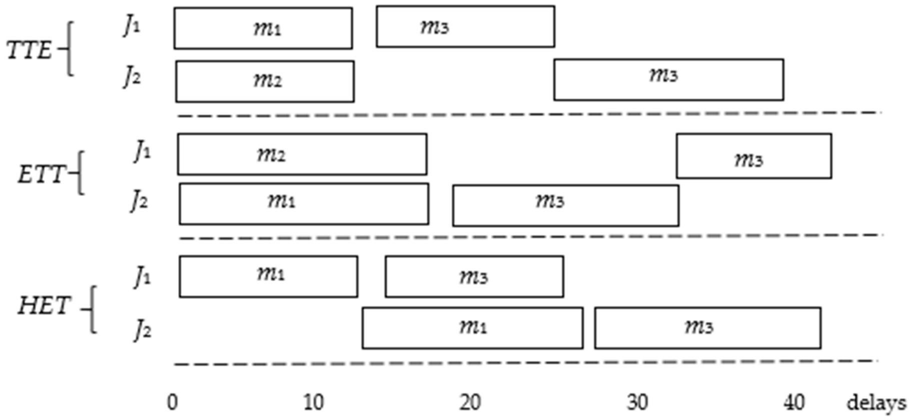

Figure 8. GANTT diagrams of the different heuristics.

Table 5. Comparison of performances between heuristic and priority rules.

Heuristic	TTE	ETT	HET
Make-span	38	43	42
Energy	48	35	40

Given the above, it appears that the proposed heuristic produces performances that are midway between minimizing the make-span and minimizing the processing and transfer of energy. This result appears to be favored by two conditions: on the one hand, the existence of an inverse relationship between time and energy required on two machines capable of carrying out the same operation (the faster machine expending more energy), and, on the other hand, if the structure of the production plan induces between the machines of the job shop, a rank in accordance with the solution building process of the heuristic.

5.3. Illustration of Resulting Scheduling in a Flexible Job Shop

The number of jobs n and the number of machines m are equal to 7, and the job shop is organized into three rows of machines to process raw material (job 1 and job 2), semi-raw material (job 3, job 4, and job 5), and semi-finished products (job 6 and job 7). The machines of the job shop and the possible connections between them are represented in Figure 9. There, jobs 1 and 2 have three operations; jobs 3, 4, and 5 have two operations; and jobs 6 and 7 have one operation. To these operations, a dummy initial and a dummy final operation are added for each job. For job 1, operation O_{12} can be performed by machines 1 and 2, operation O_{13} can be performed by machines 3 or 4, and operation O_{14} can be performed by machine 6. For job 2, operation O_{22} can be performed by machines 1 and 2, operation O_{23} can be performed by machines 3 or 4, and operation O_{24} can be performed by machine 7. For job 3, operation O_{32} can be performed by machine 4, and operation O_{33} can be performed by machines 6 or 7. For job 4, operation O_{42} can be performed by machine 3, and operation O_{43} can be performed by machines 6 or 7. For job 5, operation O_{52} can be performed by machine 5, and operation O_{53} can be performed by machine 6 or 7. For job 6, operation O_{62} can be performed by machine 6. For job 7, operation O_{72} can be performed by machine 7. This shows that there is flexibility in the assignment of machines to some operations of each job.

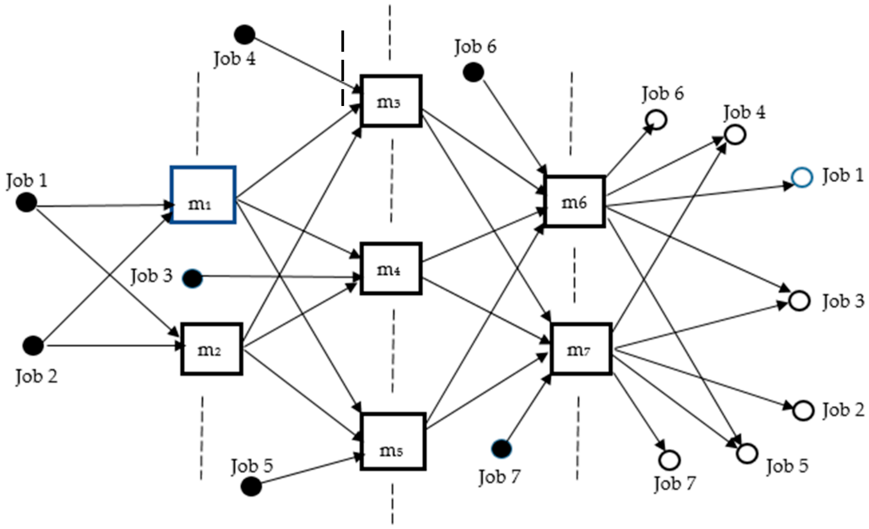

Figure 9. The considered job shop.

The following tables provide data on the operations' processing times and energy per machine (Table 6), the transportation delays and energy between machines (Table 7), and finally, the release and due times (Table 8).

Table 6. Processing delays/energy per jobs and machines.

Machines	$\mathbf{m_1}$	$\mathbf{m_2}$	$\mathbf{m_3}$	$\mathbf{m_4}$	$\mathbf{m_5}$	$\mathbf{m_6}$	$\mathbf{m_7}$
Job 1	8/50	12/40	15/70	15/60	---	10/50	---
Job 2	10/55	7/40	---	15/70	15/55	---	8/60
Job 3	---	---	---	8/30	---	15/30	15/25
Job 4	---	---	15/50	---	---	10/40	8/45
Job 5	---	---	---	---	10/60	6/60	10/50
Job 6	---	---	---	---	---	12/50	---
Job 7	---	---	---	---	---	---	15/55

Table 7. Transportation delays and energy.

Delay/Energy	$\mathbf{m_1}$	$\mathbf{m_2}$	$\mathbf{m_3}$	$\mathbf{m_4}$	$\mathbf{m_5}$	$\mathbf{m_6}$	$\mathbf{m_7}$
m_1	---	---	5/5	5/5	8/10	---	---
m_2	---	---	8/10	5/5	5/5	---	---
m_3	---	---	---	---	---	5/15	8/10
m_4	---	---	---	---	---	5/5	5/5
m_5	---	---	---	---	---	8/10	5/5
m_6, m_7	---	---	---	---	---	---	---

Table 8. Release and due times per jobs.

Jobs	**Job 1**	**Job 2**	**Job 3**	**Job 4**	**Job 5**	**Job 6**	**Job 7**
Release time	0	15	10	0	10	15	20
Due time	65	65	55	55	55	45	45

Here, only the delays and energy consumption resulting from the transfer of products between machines are considered; they are also supposed to be independent of the type of product that is transported.

The due times were chosen so that the jobs requiring only final machining (jobs 6 and 7) are cleared first to free up space in the job shop for the remaining activities of the other jobs. This is repeated with jobs 3, 4, and 5 concerning jobs 1 and 2.

5.4. Comparison of Results of the Heuristic Concerning the Priority Rules

The results obtained with the proposed heuristic in Section 4 are compared with the results of the priority rules TTE and ETT. When due times are not considered, the range of values for the necessary energy is [725, 785]. The heuristic presented in Section 4 provides a solution after 18 iterations; the obtained scheduling is displayed in Figure 10.

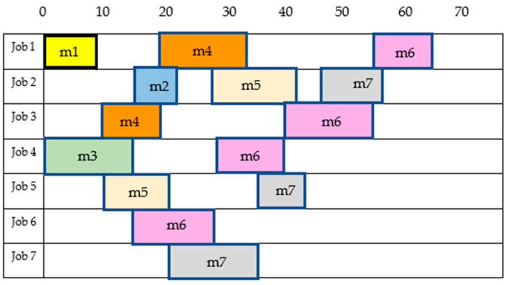

Figure 10. Gantt diagram of HET heuristic solution.

This solution has a total energy cost of 725 with a make-span of 65, while the due times are respected for each job.

Figures 11 and 12 present the corresponding Gantt diagrams, and Table 9 presents a comparison of their performances.

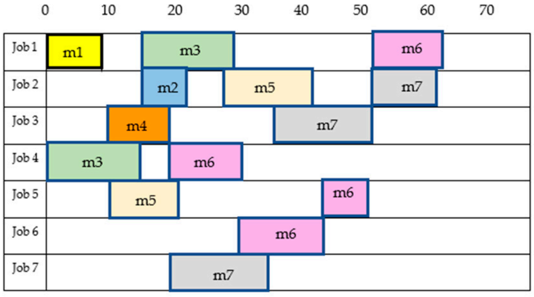

Figure 11. Gantt diagram of TTE solution.

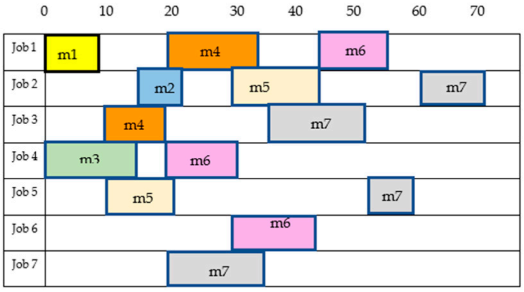

Figure 12. Gantt diagram of ETT solution.

Table 9. Performances of heuristic HET and priority rules TTE and ETT.

	Energy Cost	**Make-Span**
HET	725	65
TTE	745	62
ETT	725	70

In the considered case study, the proposed heuristic reaches the optimal performance concerning total energy, while due time constraints are satisfied for all the jobs. The TTE priority rule reduces the make-span of the considered production plan (from 65 to 62), but this is achieved at the expense of an increase in the energy cost, which is 33.3% of the total energy variation range [785,725]. Finally, the ETT priority rule achieves the same energy performance as the heuristic, but the two jobs do not comply with their due dates.

6. Conclusions

This study addressed the problem of minimizing the energy spent in a flexible production workshop with the operation of its machines and its internal logistics. The hypotheses retained led to the formulation of a particular scheduling problem that fits well with operational practice in this field. There, the operation of the flexible job shop is considered not as a mere juxtaposition of tasks but as a set of coordinated production flows that can be assigned to different machines. The field of application of this study goes well beyond the scheduling in industrial job shops and, among others, can also cover the operation of intermodal transport terminals. This is particularly important in the case of airport operations when considering the ground handling activities for aircraft at arrival and departure. There, due time constraints have an essential role in guaranteeing the punctuality of departure flights, while the energy consumption and the emissions of the numerous ground handling vehicles must be minimized as much as possible [45].

The optimization problem considered in this study was formulated as a mono-criterion optimization problem, avoiding the introduction of relative weights or the exploration of a Pareto frontier involving energy and make-span. At the same time, the consideration of release and due times allows the integration of a flexible job shop into more global processes of Industry 4.0. Its two-level structure is the result of having adopted as decision variables, besides the scheduling variables of the operations, the assignment of machines to the different operations of the jobs. This two-level structure offers opportunities to use known optimization approaches to obtain exact solutions for real-size instances of this problem. However, considering its computational complexity and the requirements to generate efficient schedules in the dynamic environment of flexible job shops, an ad hoc heuristic was designed.

With this heuristic, the machine assignments to operations are not generated a priori for each job but are constructed step by step according to the information available about the downstream operational state of the job shop. The solution generated by this heuristic derives from a permanent trade-off between energy and delays, where the subsets of candidate machines to perform an operation can be enlarged to include less energy-efficient machines that operate faster, satisfying the due time constraints for each job. This heuristic was applied with success to a small-size case study, where it outperformed two basic scheduling rules. Given the results obtained so far, it seems interesting to evaluate the performance of this heuristic when applied to higher-dimensional problems targeting other application domains.

The present research can be completed following different directions:

- Enlarging the scope of the problem by considering other sources of energy consumption (idle machines, for example).

- Enhancing the efficiency of the heuristic—for example, by integrating priority rules, such as the critical ratio (between the time remaining until the due date and the work time remaining), to better cope with the due date constraints [46].

To further examine the quality of the solutions provided to the minimum-energy flexible job-shop scheduling problem, this study can be a basis for the development of metaheuristics, such as those described in [47,48]. This will allow the use of artificial intelligence techniques to cope more globally with the search for an efficient solution.

Author Contributions: Conceptualization, O.A.O., F.L.P.K. and F.M.-C.; methodology, O.A.O., F.L.P.K. and F.M.-C.; validation, O.A.O., F.L.P.K. and F.M.-C.; investigation, O.A.O., F.L.P.K. and F.M.-C.; writing—original draft preparation, O.A.O. and F.M.-C.; writing—review and editing, F.M.-C.; visualization, O.A.O. and F.L.P.K.; supervision and project administration, F.L.P.K.; funding acquisition, O.A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Acknowledgments: The corresponding author acknowledges the special logistics support of Durban University of Technology during this research.

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

References

1. Shaardan Nur, A.; Roslin Eida, N.; Ahamat, M. Lean Management Concept in Energy Efficiency Improvement in Non-domestic. *Int. J. Appl. Eng. Res.* **2017**, *12*, 15242–15251.
2. Zhang, L.; Li, X.; Gao, L.; Zhang, G. Dynamic rescheduling in FMS that is simultaneously considering energy consumption and schedule efficiency. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 1387–1399. [CrossRef]
3. Meng, L.; Zhang, C.; Shao, X.; Ren, Y. MILP models for energy-aware flexible job shop scheduling problem. *J. Clean. Prod.* **2018**, *210*, 710–723. [CrossRef]
4. Gahm, C.; Denz, F.; Dirr, M.; Tuma, A. Energy-efficient, scheduling in manufacturing companies: A review and research framework. *Eur. J. Oper. Res.* **2015**, *248*, 744–757. [CrossRef]
5. Tang, D.; Dai, M. Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem. *Chin. J. Mech. Eng.* **2015**, *28*, 1048–1055. [CrossRef]
6. Xie, J.; Gao, L.; Peng, K.; Li, X.; Li, H. Review on flexible job shop scheduling. *IET Collab. Intell. Manuf.* **2019**, *1*, 67–77. [CrossRef]
7. Dauzière-Pérès, S.; Ding, J.; Shen, L.; Tamssaouet, K. The flexible job shop scheduling problem: A review. *Eur. J. Oper. Res.* **2024**, *314*, 409–432. [CrossRef]
8. Fattahi, P.; Mehraba, M.S.; Jolai, F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *J. Intell. Manuf.* **2007**, *18*, 331–342. [CrossRef]
9. Demir, Y.; Kursat Islayen, S. Evaluation of mathematical models for flexible job-shop scheduling. *Appl. Math. Model.* **2013**, *37*, 977–988. [CrossRef]
10. Chaudhry, I.A.; Khan, A.A. A research survey: Review of flexible job shop scheduling techniques. *Int. Trans. Oper. Res.* **2016**, *23*, 551–591. [CrossRef]
11. Coelho, P.; Pinto, A.; Moniz, S.; Silva, C. Thirty years of flexible job-shop scheduling: A bibliometric study. *Procedia Comput. Sci.* **2021**, *180*, 787–796. [CrossRef]
12. Pezzella, F.; Morganti, G.; Ciaschetti, G. A genetic algorithm for the flexible job-shop scheduling problem. *Comput. Oper. Res.* **2008**, *35*, 3202–3212. [CrossRef]
13. Zheng, Y.; Lian, L.; Mesghouni, K. Comparative study of heuristics algorithms in solving flexible job shop scheduling problem with condition based maintenance. *J. Ind. Eng. Manag.* **2014**, *7*, 518–531.
14. Perroux, T.; Arbaoui, T.; Merghem-Boulahia, L. A mathematical model for the flexible job-shop scheduling problem with availability constraints. *IFAC Pap.* **2023**, *56*, 5388–5393. [CrossRef]
15. Shen, X.N.; Han, Y.; Fu, J.Z. Robustness measures and robust scheduling for multi-objective stochastic flexible job shop scheduling problems. *Soft Comput.* **2017**, *21*, 6531–6554. [CrossRef]

16. Elgendy, A.E.; Hussein, M.; Elhakeem, A. Optimising dynamic flexible job shop problem based on genetic algorithm. *Int. J. Curr. Eng. Technol.* **2017**, *7*, 368–373.
17. Mohan, J.; Lanka, K.; Rao, A.N. A review of dynamic job shop scheduling techniques. *Procedia Manuf.* **2019**, *30*, 34–39. [CrossRef]
18. Echsler Minguillon, F.; Stricker, N. Robust predictive-reactive scheduling and its effect on machine disturbance mitigation. *CIRP Ann.-Manuf. Technol.* **2020**, *69*, 401–404. [CrossRef]
19. Zhou, T.; Zhu, H.; Tang, D.; Liu, C.; Cai, Q.; Shi, W.; Gui, Y. Reinforcement learning for online optimization of job-shop scheduling in smart manufacturing factory. *Adv. Mech. Eng.* **2022**, *14*, 1–19. [CrossRef]
20. Wu, X.; Sun, Y. A green scheduling algorithm for flexible job shop with energy-saving measures. *J. Clean. Prod.* **2018**, *172*, 3249–3264. [CrossRef]
21. Gao, K.; Huang, Y.; Sadollah, A.; Wang, L. A review of energy-efficient scheduling in intelligent production systems. *Complex Intell. Syst.* **2020**, *6*, 237–249. [CrossRef]
22. He, Y.; Li, Y.; Wu, T.; Sutherland, J.W. An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining shops. *J. Clean. Prod.* **2014**, *87*, 245–254. [CrossRef]
23. May, G.; Stah, B.; Taisch, M.; Prabhu, V. Multi-objective genetic algorithm for energy-efficient job shop scheduling. *Int. J. Prod. Res.* **2015**, *53*, 7071–7089. [CrossRef]
24. Liu, Y.; Dong, H.; Lohs, N.; Petrovic, S. A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. *Int. J. Prod. Econ.* **2016**, *179*, 259–272. [CrossRef]
25. Mokhta, H.; Hasani, A. An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Comput. Chem. Eng.* **2017**, *104*, 339–352. [CrossRef]
26. Zhang, Z.; Wu, L.; Peng, T.; Jia, S. An Improved Scheduling Approach for \Minimizing Total Energy Consumption and Makespan in Flexible Job Shop Environment. *Sustainability* **2019**, *11*, 179. [CrossRef]
27. Salido, M.A.; Escamilla, J.; Barber, F.; Giret, A.; Tang, D.; Dai, M. Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems. *AI EDAM* **2016**, *30*, 300–321. [CrossRef]
28. Rakovitis, N.; Li, D.; Zhang, N.; Li, J.; Zhang LAnd Xiao, X. Novel approach to energy-efficient flexible job-shop scheduling problems. *Energy* **2022**, *238*, 121773. [CrossRef]
29. Nouiri, M.; Bekar, A.; Trentesaux, D. Towards energy efficient scheduling and rescheduling for dynamic flexible job shop problem. *IFAC Pap.* **2018**, *51*, 1275–1280. [CrossRef]
30. Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. *Ann. Oper. Res.* **1993**, *41*, 157–183. [CrossRef]
31. Gendreau, M. An Introduction to Tabu Search. In *Handbook of Metaheuristics. International Series in Operations Research & Management Science*; Glover, F., Kochenberger, G.A., Eds.; Springer: Boston, MA, USA, 2003; Volume 57.
32. Lenstra, J.K.; Rinnooy Kan AH, G.; Brucker, P. Complexity of machine scheduling problems. In *Annals of Discrete Mathematics*; Elsevier: Amsterdam, The Netherlands, 1977; Volume 1, pp. 343–362.
33. Olanrewaju, O.A.; Kabuya, K.; Ramirez, L.; Mora-Camino, F. A Model-Based Heuristic for Minimum Energy Scheduling of Flexible Job-Shop Programs. In Proceedings of the IEEE CoDIT 2024 Conference, Valetta, Malta, 1–4 July 2024.
34. Rahmaniani, R.; Crainic, T.G.; Gendreau, M.; Rei, W. The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* **2017**, *259*, 801–817. [CrossRef]
35. Kalashnikov, V.A.; Dempe, S.; Pérez-Valdés, G.; Kalashnykova, N.; Camacho-Vallejo, J.C. Bilevel programming and applications. *Math. Probl. Eng.* **2015**, *2015*, 310301. [CrossRef]
36. Dempe, S.; Zemkoho, A. Springer Optimization and Its Applications. In *Bilevel optimization Advances and New Challenges*; Pardalos, P.M., Thai, M.T., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 161.
37. Angelo, J.S.; Barbosa HJ, C. A study on the use of heuristics to solve a bilevel programming problem. *Int. Trans. Oper. Res.* **2015**, *22*, 861–882. [CrossRef]
38. Camacho-Vallejo, J.F.; Corpus, C.; Villegas, J.G. Metaheuristics for bilevel optimization: A comprehensive review. *Comput. Oper. Res.* **2024**, *161*, 106410. [CrossRef]
39. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; ISBN 978-0-387-31073-2.
40. Aytug, H.; Bhattacharyya, S.; Koehler, G.J.; Snowdon, J.L. A review of machine learning in scheduling. *IEEE Trans. Eng. Manag.* **1994**, *41*, 165–171. [CrossRef]
41. Behnke, D.; Geiger, M.J. *Test Instances for Job Shop Scheduling Problems with Work Centers*; Research Report RR-12-01-01; Helmut Schmit University: Hamburg, Germany, 2012.
42. Araujo KA, G.; Birgin, E.G.; Ronconi, D.P. Models, constructive heuristics, and benchmark instances for the flexible job shop scheduling problem with sequencing flexibility and position-based learning effect. *arXiv* **2024**, arXiv:2403.16766.
43. Haupt, R. A survey of priority rule-based scheduling. *OR Spektrum* **1989**, *11*, 3–16. [CrossRef]
44. Demir, Y.; Yilmaz, H. An efficient priority rule for flexible job shop scheduling problem. *J. Eng. Res. Appl. Sci.* **2021**, *10*, 1906–1918.
45. Alonso Tabares, D.; Olanrewaju, O.A.; Krykhtine, F.P.; Felix Mora-Camino, F. Characterizing Airport Ground Handling as a Multi Flexible Flow Shop. In Proceedings of the XXI SITRAER Conference-SBPTA, Fortaleza, Brazil, 16–18 October 2024.
46. Lödding, H.; Piontek, A. The Surprising Effectiveness of Earliest Operation Due-date Sequencing. *Prod. Plan. Control.* **2017**, *28*, 459–471. [CrossRef]

47. Abdolrazzagh-Nezhad, A.; Abdullah, S. A Review on Metaheuristic Approaches for Job-Shop Scheduling Problems. *Data Sci. J. Comput. Appl. Inform.* **2024**, *8*, 45–63. [CrossRef]
48. Fuladi, S.K.; Kim, C.-S. Dynamic Events in the Flexible Job-Shop Scheduling Problem: Rescheduling with a Hybrid Metaheuristic Algorithm. *Algorithms* **2024**, *17*, 142. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.