Minimizing Energy Cost in a Job-Shop Scheduling Problem Under ToU Pricing: A New Method Based on a Period-Indexed MILP

Marouane Felloussi[a], Xavier Delorme[b] and Paolo Gianessi[c]

Mines Saint-Etienne, Univ. Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, F-42023 Saint-Etienne, France

{*marouane.felloussi, delorme, paolo.gianessi*}*@emse.fr*

Keywords: Energy-Efficient Production Systems, ToU Pricing, Job-Shop Scheduling, Exact Method, Mixed-Integer Linear Programming, Valid Inequalities.

Abstract: This work addresses the job-shop scheduling problem under energy considerations, specifically focusing on minimizing total energy costs within a Time-of-Use pricing framework, denoted as $J_m||TEC$. We propose a period-indexed Mixed-Integer Linear Programming formulation, which proves advantageous due to its smaller model size compared to traditional time-indexed approaches. Initial studies highlight that while our model can rapidly find feasible solutions, it struggles with weak linear relaxations. Different families of valid inequalities are thus considered to improve the obtained lower bounds. In order to evaluate and compare the impact of the proposed valid inequalities, computational experiments are presented and numerical results are discussed and analyzed.

1 INTRODUCTION

The rising demand for energy, volatile prices, and global warming concerns have heightened the focus on energy efficiency. The industrial sector, responsible for over half of global end-use energy consumption, is projected to see electricity use grow from 22% in 2021 to 46% by 2050 due to process electrification (International Energy Agency, 2023). Manufacturing systems, as key contributors to industrial energy use, must meet the rising demand for goods while addressing their energy-intensive nature. Enhancing energy efficiency in this sector is crucial for sustainability. Energy cost minimization provides a strategic means of improving efficiency by leveraging *Time-of-Use (ToU) pricing*, where varying per-unit energy prices incentivize production during off-peak periods. Such scheduling not only reduces costs and grid strain, but also lowers greenhouse gas emissions.

As a relevant problem in manufacturing systems, the *Job-Shop Scheduling Problem* (JSSP) stands out as one of the most complex and studied problems in operations research. The $\mathcal{NP}$-hard (Garey et al., 1976) problem consists of operations that must be contiguously processed on dedicated machines and in a predefined order. Each operation has a specific machine that it needs to be processed on and only one operation in a job can be processed at a given time. The problem then consists in sequencing the operations on machines in order to minimize the makespan, i.e. the last completion time.

Incorporating energy considerations into the JSSP adds complexity, even with simplified assumptions like constant machine power usage. For instance, when energy prices follow a ToU pricing scheme, minimizing energy costs without worsening production objectives results in a scheduling problem that must optimize both production and energy efficiency.

A key combinatorial challenge in energy-efficient scheduling is that incorporating non-regular criteria like time-dependent energy pricing requires explicit timing decisions (Dauzère-Pérès et al., 2024). To model time-dependent energy costs, time-indexed (TI) formulations are often preferred to disjunctive (D) ones (see, e.g. Masmoudi et al. 2019). While TI formulations provide strong dual bounds, they often lead to large *Mixed-Integer Linear Programs* (MILP). In the most common case where ToU profiles are piecewise constant, period-indexed formulations can be an alternative modeling approach. These involve associating a variable with whether the processing of an operation occurs during one of the ToU periods.

[a] https://orcid.org/0009-0005-1351-5403
[b] https://orcid.org/0000-0002-9465-1255
[c] https://orcid.org/0000-0001-5457-3689

Felloussi, M., Delorme, X. and Gianessi, P.
Minimizing Energy Cost in a Job-Shop Scheduling Problem Under ToU Pricing: A New Method Based on a Period-Indexed MILP.
DOI: 10.5220/0013181200003893
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 14th International Conference on Operations Research and Enterprise Systems (ICORES 2025)*, pages 320-327
ISBN: 978-989-758-732-0; ISSN: 2184-4372
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

In this paper, we address the job-shop scheduling problem with minimization of the total energy costs subject to a ToU pricing scheme. To do so, we propose a novel period-indexed MILP, where timing decisions are indexed on ToU periods. Valid inequalities and variable-fixing relations are introduced and discussed to strengthen the linear relaxations.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature. Section 3 introduces key notations, presents the proposed mathematical formulation and discusses the valid inequalities used to enhance it. Computational experiments are detailed and analyzed in Section 4. We conclude the paper with remarks and potential research directions in Section 5.

2 SHORT LITERATURE REVIEW

Energy cost minimization under ToU pricing has been explored in various scheduling settings. For parallel machines, Gaggero et al. (2023) present a MILP with symmetry-breaking properties and a heuristic for a multi-objective problem addressing both energy cost and makespan. Further, Tian and Zheng (2024) tackle a single machine batch scheduling problem, developing a set partitioning formulation and novel branching rules for their branch-and-price. In the shop scheduling context, Park and Ham (2022) develop a Constraint Programming (CP) model for a flexible job-shop problem considering machine states, while Jiang and Wang (2020) introduce a time-indexed MILP for the permutation flow-shop. Some research focuses on minimizing total energy cost subject to operational constraints. Masmoudi et al. (2019) develop disjunctive and time-indexed formulations, as well as a matheuristic for the job-shop scheduling under makespan and power limits. Bley and Linß (2022) approach the job-shop with machine states and release/due dates with a branch-and-bound algorithm based on a time-indexed formulation, enhanced by preprocessing and propagation techniques.

To conclude this brief literature review, we highlight some of the works using a period-based approach. In the machine scheduling environment, Cheng et al. (2016) examine a single machine batch scheduling problem minimizing total energy cost, comparing time-indexed and period-indexed formulations. As for unrelated parallel machines, Che et al. (2017) propose a model to minimize total energy cost, enhanced with valid inequalities and a two-stage heuristic. For the same problem, Ding et al. (2016) present a period-indexed formulation and a heuristic based on a Dantzig-Wolf Reformulation (DWR). On the other hand, there is limited research on shop scheduling with period-indexed formulations. To the best of our knowledge, Ho et al. (2022) is the only work to date featuring this modeling choice, featuring a Logic-Based Benders Decomposition (LBBD) for the two-machines permutation flow-shop problem.

Table 1 formally classifies the different discussed problem settings, using the 3-field notation proposed in Graham et al. (1977) to characterize the general setting of these problems. Additionally, $\phi_{\max}$ will denote the contracted power limit, on/off the machines regulation via the “on/off” mechanism, and TEC the total energy cost of a schedule. We also adopt the naming conventions of Pinedo (2016) for scheduling problems. For MILP approaches, we specify the nature of the formulation using the previously introduced acronyms D, TI, and PI.

3 PROBLEM DESCRIPTION AND FORMULATION

3.1 Formal Problem Definition

We consider a shop floor with a set $\mathcal{M}$ of *machines* and a set $\mathcal{J}$ of *jobs* to be performed over a time horizon C. Each job $j \in \mathcal{J}$ must be executed on a predefined ordered subset of machines $M_j \subseteq \mathcal{M}$, giving rise to as many *operations* subject to precedence constraints. Without loss of generality, we consider in this study that every job executes over all machines: an operation can hence be denoted as $(j,m) \in O$ with $O := \mathcal{J} \times \mathcal{M}$ denoting the set of operations, and $|M_j| = |\mathcal{M}|, \forall j \in \mathcal{J}$. For each $j \in \mathcal{J}$, we denote $(j,m) \prec (j,m')$ if operation (j,m) must be processed before operation (j,m'), i.e. if in the ordered subset M_j, m precedes m'.

The processing time $p_{j,m}$ of an operation is known and is deterministic. Each machine m has a constant nominal power φ_m, therefore each operation (j,m) has energy consumption of $\varphi_m p_{j,m}$. Energy usage is subject to a ToU pricing, hence the time horizon C is divided into a set $\mathcal{K}$ of periods, each period $k \in \mathcal{K}$ featuring a duration l^k and a price per energy unit c^k. Let t^k denote the start date of period k, then $t^1 = 0$, $t^{k+1} - t^k = l^k, \forall k \in [\![1, |\mathcal{K}|]\!]$, $t^{|\mathcal{K}+1|} = C$.

In addition to being precedence-compliant, the processing of operations must be non-preemptive (i.e. uninterrupted) and avoid overlapping on the same machine. The goal is to find a feasible schedule of all operations within the time horizon C, such that the cost of the energy used to process them, determined by the ToU fees, is minimized. Using the three-field notation

Table 1: A summary of the cited works.

Problem class	Article	Problem	Solution Approach
job-shop scheduling	(Bley and Linß, 2022) (Masmoudi et al., 2019)	$Jm\|\text{on/off}, r_j, d_j\|TEC$ $Jm\|\phi_{\max}\|TEC$	MILP (TI), B&B MILP (D,TI), MH
flexible job-shop scheduling	(Park and Ham, 2022) (Jiang and Wang, 2020)	$FJm\|\text{on/off}\|C_{\max}, TEC$ $FJm\|\|C_{\max}, TEC$	MILP (TI), CP MILP (TI), H
flow-shop scheduling	(Ho et al., 2022)	$F2\|prmu, \text{on/off}\|TEC$	MILP (PI), LBBD
parallel machine scheduling	(Gaggero et al., 2023) (Che et al., 2017) (Ding et al., 2016)	$Pm\|\|TEC, C_{\max}$ $Rm\|\|TEC$ $Rm\|\|TEC$	MILP (TI), H MILP (PI), H MILP (PI), DWR
single machine scheduling	(Tian and Zheng, 2024) (Cheng et al., 2016)	$1\|batch\|TEC$ $1\|batch\|TEC$	CG-H MILP (PI)

CG-H : column generation based heuristic
MH : matheuristic
H : problem specific heuristic
TI : time-indexed formulation
D : disjunctive formulation
PI : period-indexed formulation

previously recalled, the problem can be referred to as $Jm||TEC$.

3.2 Mathematical Formulation

In this section, a novel MILP for the studied problem is presented. Its main feature is to be indexed on ToU periods, since the variables that allow to compute the cost of the energy consumption associated with an operation are indexed on the ToU periods. In the following, symbols $j, j' \in \mathcal{J}$, $m, m' \in \mathcal{M}$ and $k, k' \in \mathcal{K}$ will respectively denote jobs, machines and ToU periods. The decision variables are: **1)** binary *processing status* variables $x^k_{j,m}$, modeling whether operation (j,m) is processed during period k or not, **2)** non-negative *processing duration* variables $d^k_{j,m}$, equal to the duration of the processing of operation (j,m) in period k, **3)** non-negative variables $s_{j,m} \in [0, C - p_{j,m}]$ and $c_{j,m} \in [p_{j,m}, C]$, denoting the *starting* and *completion dates* of operation (j,m) – also known in the literature as *natural date variables*, **4)** binary *machine disjunction* variables $u_{j,j',m}$ for $j < j'$, equal to 1 if job j is processed earlier than j' on machine m, and to 0 if it is processed later than j' on machine m. Further, we define for each operation (j,m) the values

$$\underline{s}_{j,m} = \sum_{\substack{m' \in O_j \\ (j,m') \prec (j,m)}} p_{j,m'}, \quad \overline{s}_{j,m} = C - p_{j,m} - \sum_{\substack{m' \in O_j \\ (j,m) \prec (j,m')}} p_{j,m'} \tag{1}$$

as the earliest and latest starting dates respectively. $\underline{c}_{j,m}$ and $\overline{c}_{j,m}$ are defined similarly.

Objective Function: The objective function minimizing the total operational cost of a schedule is

$$\min \sum_{k \in \mathcal{K}} c^k \sum_{m \in M} \varphi_m \sum_{j \in \mathcal{J}} d^k_{j,m}, \tag{2}$$

which is the sum of energy consumption scaled by the price over the corresponding periods. The constraints in the proposed MILP are as follows:

Processing Times Partition: To ensure that the operations are entirely executed within the given makespan C, we partition the processing times for each operation (j,m) across the different periods:

$$\sum_k d^k_{j,m} = p_{j,m}, \quad \forall (j,m) \in O. \tag{3}$$

Non-Preemption: To guarantee that operations are not preempted, it suffices to link the processing time variables to the natural date ones, which we also need for precedence. First, variables x and d are linked through the constraint

$$d^k_{j,m} \leqslant \min(l^k, p_{j,m}) x^k_{j,m}, \quad \forall (j,m) \in O, \forall k \in \mathcal{K}, \tag{4}$$

and to the natural date variables through the following constraints:

$$\begin{cases} d^k_{j,m} \leqslant t^{k+1} - s_{j,m} + \overline{s}_{j,m}(1 - x^k_{j,m}), \\ d^k_{j,m} \leqslant c_{j,m} - t^k x^k_{j,m}. \end{cases} \forall (j,m) \in O, \forall k \in \mathcal{K}, \tag{5}$$

Constraints (4) act jointly with constraints (5) :

- If $x^k_{j,m} = 1$, i.e. operation (j,m) executes on period k, then (j,m) has began executing in that period or prior to it. The first period is the smallest k' for which $x^{k'}_{j,m} = 1$. It has the smallest gap $t^{k'+1} - s_{j,m}$, thus $d^k_{j,m} \leqslant t^{k'+1} - s_{j,m}$.
- The second constraint is similar but involves the end dates: the operation completes in the current period k or after it. The last period is the largest k' for which $x^{k'}_{j,m} = 1$. It has the smallest gap $c_{j,m} - t^k$, thus $d^k_{j,m} \leqslant c_{j,m} - t^{k'}$.

- In case operation (j,m) starts and ends in the same period k, both relations (5) are dominated by the equation $d^k_{j,m} \leqslant p_{j,m}$, imposed by (4).
- if (j,m) spans over $n \geqslant 2$ consecutive periods $[\![k, k+n-1]\!]$, the constraints (4) associated with k and $k+n-1$ are dominated, respectively, by the associated constraints in (5), while the opposite occurs for intermediate periods $[\![k+1, k+n-2]\!]$.
- The first constraint of (5) is not binding if $x^k_{j,m} = 0$, whereas the second constraint is not binding if $x^k_{j,m} = 0$ due to (4).

Further, the following constraint links the natural date variables of an operation:

$$c_{j,m} = s_{j,m} + p_{j,m}, \quad \forall (j,m) \in \mathcal{O}, \tag{6}$$

ensuring that an operation (j,m) is processed without preemption, whereas relations (3), (4) and (5) guarantee that the value of its processing time $p_{j,m}$ is correctly partitioned into consecutive ToU periods.

Precedence: Precedence constraints write as

$$c_{j,m} \leqslant s_{j,m'}, \quad \forall j \in \mathcal{J}, \forall m, m' \in M_j : (j,m) \prec (j,m'), \tag{7}$$

i.e. (j,m) completes before the start of (j,m'). Note that $c_{j,m}$ could be replaced by $s_{j,m} + p_{j,m}$ in the different equations. We maintain this notation for clarity.

Machine Disjunction: To ensure that any two given operations scheduled on the same machine do not overlap, we define the precedence variables through the following *big-M* constraints:

$$\begin{cases} c_{j,m} - s_{j',m} \leqslant \Delta_{j,j',m}(1 - u_{j,j',m}), & \forall j, j' \in \mathcal{J} : j < j' \\ c_{j',m} - s_{j,m} \leqslant \Delta_{j',j,m}\, u_{j,j',m}, & \forall m \in \mathcal{M}, \end{cases} \tag{8}$$

where $\Delta_{j,j',m} := \overline{c}_{j,m} - \underline{s}_{j',m}$. If j precedes j' on machine m, i.e. $c_{j,m} \leqslant s_{j',m}$, then $u_{j,j',m} = 1$.

3.3 Valid Inequalities

The introduced model PI is compact, involving a polynomial number of variables and constraints. However, due to the big-M constraints that linearize its disjunctive structure, it can result in weak linear relaxations despite using tight bounds on these constraints (Conforti et al., 2014). In the following section, we introduce different families of valid inequalities aimed at improving the introduced model.

Transitive Precedence: If operation (j,m) is scheduled on machine m before (j',m), and (j',m) is scheduled before (j'',m), then $u_{j,j'',m}$ must take value 1:

$$u_{j,j',m} + u_{j',j'',m} - 1 \leqslant u_{j,j'',m}, \quad \begin{array}{r} \forall j, j', j'' \in \mathcal{J} : j < j' < j'' \\ \forall m \in \mathcal{M}. \end{array} \tag{9}$$

Consecutive Period Processing Inequalities: Given a machine m and a period k, at most one operation (j,m) can be processed over k and the following period $k+1$, yielding in that case $x^k_{j,m} + x^{k+1}_{j,m} = 2$. Based on this consideration, we can write:

$$x^k_{j,m} + x^{k+1}_{j,m} + x^k_{j',m} + x^{k+1}_{j',m} \leqslant 3, \quad \begin{array}{r} \forall j, j' \in \mathcal{J} : j < j', \\ \forall m \in \mathcal{M}, \forall k \leqslant |\mathcal{K}| - 1. \end{array} \tag{10}$$

Via similar reasoning, all operations (j,m) of duration lesser or equal than the shortest ToU period can execute over at most two periods:

$$\sum_{k \in \mathcal{K}} x^k_{j,m} \leqslant 2, \quad \forall (j,m) \in \mathcal{O} : p_{j,m} \leqslant \min_k l^k. \tag{11}$$

Non-Preemption Inequalities: If operation (j,m) completes at a period k, i.e. $x^k_{j,m} = 1$ and $x^{k+1}_{j,m} = 0$, then processing cannot occur at periods $k' \geqslant k+1$:

$$\sum_{k'=k+1}^{|\mathcal{K}|} x^{k'}_{j,m} \leqslant (|\mathcal{K}| - k)(1 - x^k_{j,m} + x^{k+1}_{j,m}), \quad \begin{array}{r} \forall (j,m) \in \mathcal{O}, \\ \forall k \leqslant |\mathcal{K}| - 1. \end{array} \tag{12}$$

Similarly, we can derive inequalities for the opposite case, in which the processing of (j,m) starts at k:

$$\sum_{k'=1}^{k-2} x^{k'}_{j,m} \leqslant (k-2)(1 - x^k_{j,m} + x^{k-1}_{j,m}), \quad \begin{array}{r} \forall (j,m) \in \mathcal{O}, \\ \forall k \geqslant 3. \end{array} \tag{13}$$

Further, if the processing of (j,m) extends over two non-consecutive periods, then it must process over the intermediate one:

$$x^k_{j,m} \geqslant x^{k-1}_{j,m} + x^{k+1}_{j,m} - 1, \quad \begin{array}{r} \forall (j,m) \in \mathcal{O}, \\ \forall k \in [\![2, |\mathcal{K}| - 1]\!]. \end{array} \tag{14}$$

Precedence Inequalities: Consider two operations of the same job (j,m) and (j,m') s.t. $(j,m) \prec (j,m')$. If (j,m') executes over period k, (j,m) cannot execute over all subsequent periods. As such, we have:

$$\sum_{k'=k+1}^{K} x^{k'}_{j,m} \leqslant (|\mathcal{K}| - k)(1 - x^k_{j,m'}), \quad \begin{array}{r} \forall m, m' \in O_j : \\ (j,m) \prec (j,m'), \\ \forall k \leqslant |K| - 1. \end{array} \tag{15}$$

The explanation of (15) is similar to that of relations (12) and (13). They are non-binding if $x^k_{j,m'} = 0$, and force all variables $x^{k'}_{j,m}, (k' > k)$ to be 0 otherwise. By the same logic, this applies to the preceding periods:

$$\sum_{k'=1}^{k-1} x^{k'}_{j,m'} \leqslant (k-1)(1 - x^k_{j,m}), \quad \begin{array}{r} \forall m, m' \in O_j : \\ (j,m) \prec (j,m'), \\ \forall k \geqslant 2. \end{array} \tag{16}$$

Processing Duration Inequalities: We can derive upper bounds on period processing duration, i.e. $d^k_{j,m} \leqslant \overline{d}^k_{j,m}$, where for each operation (j,m) and period

k, we have:

$$\overline{d}_{j,m}^k = \begin{cases} 0 & \text{if } t^{k+1} \leqslant \underline{s}_{j,m} \vee \overline{c}_{j,m} \leqslant t^k, \\ t^{k+1} - \underline{s}_{j,m} & \text{if } t^k \leqslant \underline{s}_{j,m} \leqslant t^{k+1}, \\ \overline{c}_{j,m} - t^k & \text{if } t^k \leqslant \overline{c}_{j,m} \leqslant t^{k+1}, \\ l^k & \text{otherwise.} \end{cases}$$

This allows us to replace (4) by a stronger inequality:

$$d_{j,m}^k \leqslant \min(l_k, p_{j,m}, \overline{d}_{j,m}^k) x_{j,m}^k, \quad \forall (j,m) \in \mathcal{O}, \forall k \in \mathcal{K}. \tag{17}$$

Non-Overlap Inequalities: Given a period k and a machine m, the sum of the processing durations of all of the associated operations must not exceed the period length due to the non-overlapping requirement. This also applies to operations of jobs given the precedence relations. It follows that:

$$\sum_{j \in \mathcal{J}} d_{j,m}^k \leqslant l^k, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{K}. \tag{18}$$

$$\sum_{m \in \mathcal{M}} d_{j,m}^k \leqslant l^k, \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}. \tag{19}$$

Consider the feasible schedule provided in Figure 1 with jobs J_1 and J_2 and machines M_1, M_2 and M_3. In the second period starting at t^1 and ending at t^2, we have $d_{1,2}^2 + d_{2,2}^2 \leqslant l^2$ and $d_{2,2}^2 + d_{2,1}^2 + d_{2,3}^2 \leqslant l^2$.

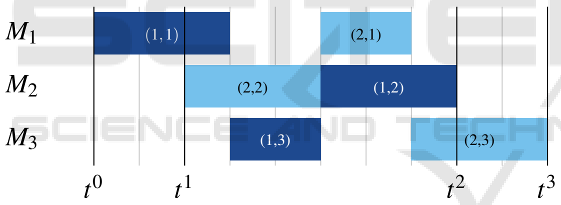

Figure 1: A two job, three machine example.

Note: Inequalities similar to (12), (13) and (18) were used as model-defining constraints in Che et al. (2017) for a parallel machine scheduling problem.

4 COMPUTATIONAL EXPERIMENTS

In this section, we analyze the performance of the time-indexed model against the proposed period-indexed one. Further, we seek to assess the impact of the derived valid inequalities on improving the model linear relaxation and the Branch&Bound tree search. The MILPs have been implemented on Julia 1.10 and solved using Gurobi 11.0. Experiments are run with a time limit of one hour, on a single thread of a 28-core Intel Xeon Gold 6132, 2.60 GHz machine.

Set of Instances: We tested our formulation on the benchmark instances provided by Masmoudi et al. (2019). In that work, the $Jm|\phi_{\max}|TEC$ is tackled, which generalizes the problem presented here by considering a limit on the power usage along the schedule, and two families of instances are derived from the JSSP instances ft06 (Fisher and Thompson, 1963) with 6 jobs and 6 machines, and la04 (Lawrence, 1984) with 5 jobs and 10 machines. The authors enhance the instances with: **a)** 5 different sets φ_i of machine power values drawn randomly from $\mathcal{U}[5,10]$, **b)** 3 different planning horizon values $\lambda \cdot C^\star$ with $\lambda \in \{1.0, 1.1, 1.2\}$, $C^\star$ denoting the minimal makespan under a maximum power constraint, **c)** 3 different power peak thresholds equal to $\alpha \cdot \sum_m \varphi_m$ with $\alpha \in \{0.7, 0.9, 1.0\}$. Of the 90 resulting instances, we consider the 30 having $\alpha = 1.0$, as this makes the peak power constraint non-binding and gives rise to the $Jm||TEC$ problem studied here. As for the ToU tariffs, we considered the ones used in the original instances based on the electricity price profiles in France with on- and off-peak periods.

Base Model: In Table 2, we compare the results of the reference time-indexed model of Masmoudi et al. (2019) referred to as IP2, and the period-indexed formulation (3)-(8), which we refer to as PI_0. The comparison is performed on the basis of the 30 considered instances under a 3600s time limit, averaged over the machine power values, since they are drawn from the same uniform distribution. For each model, the column T/(%)/[#$\overline{\mathbf{x}}$] reports the CPU time T for solving the instance to optimality, or the final gap (%) if optimality is not proven within the time limit alongside the number of feasible solutions [#$\overline{\mathbf{x}}$]. We also report the gap $\%^r$ between the root relaxation and the best known solution, as well as the number of nodes #n explored in the B&B tree before optimality is proven or the time limit is exceeded. The best known solution is always optimal, as for all instances, at least one of the two algorithms finds an optimum within time limit. Further, we provide a comparison of the size of the two models using the ratio of the number of variables (#cols) and that of the constraints (#rows), both measured prior to Gurobi presolve.

The results seem to suggest that the parameter λ has an impact on the effectiveness of IP2 and PI_0. In fact, for $\lambda = 1.0$, the PI_0-based algorithm consistently outperforms its IP2 counterpart despite larger root gaps: optimal solutions are found significantly faster for the 5 ft06 instances, while for the la04 ones, for which IP2 cannot find cannot prove optimality (2 cases) or even find feasible solutions (3 cases), PI_0 converges to optimal solutions in 20s on average.

In general, the PI_0-based algorithm is capable of finding feasible solutions more quickly while IP2

Table 2: Comparison of the proposed model in its base form PI_0 and the time-indexed model IP2 of Masmoudi et al. (2019).

	IP2			PI_0			relative $\left(\frac{PI_0}{IP2}\right)$	
inst./λ	T/(%)/[#$\bar{x}$]	$\%^r$	#n	T/(%)/[#$\bar{x}$]	$\%^r$	#n	#cols	#rows
ft06/1.0	0.25	0.27%	1	0.02	12.37%	1	0.20	0.33
ft06/1.1	0.368	0.05%	1	2.21	10.68%	2222	0.19	0.30
ft06/1.2	2.842	0.17%	16.4	64.40	10.28%	51K	0.20	0.32
la04/1.0	(0.63%)/[2]	0.47%	60	20.626	12.60%	3910	0.023	0.04
la04/1.1	1432.06	0.00%	8.6	(2.82%)/[5]	11.85%	612K	0.024	0.04
la04/1.2	2564.31	0.00%	27.6	(5.25%)/[5]	11.85%	506K	0.022	0.04

yields a tighter linear relaxation, which is in line with the conclusions from Masmoudi et al. (2019). For $\lambda = 1.0$, PI_0 is able to reach optimality within the time limit as a result of a reduced search space, due to the scheduling horizon being the smallest possible. In contrast, the IP2-based algorithm encounters difficulties in finding initial feasible solutions, particularly for the la04 instances. The issue with IP2 appears to be the computational time required for linear relaxations at each node of the B&B tree. Specifically, for the la04 set at $\lambda = 1.0$, at most 60 nodes are visited on average. Meanwhile, for PI_0, the linear relaxation of a node is computed much more rapidly, allowing it to explore more nodes.

As λ increases beyond 1.0, the longer time horizons expand the search space and expose the symmetry of PI_0, leading to a shift in performance between the two models. Despite the increased time required to prove optimality, the IP2-based algorithm finds feasible solutions more easily than PI_0 on the la04 set and, with its tighter dual bound, proves optimality more quickly after exploring a comparable number of nodes. Particularly, it can find the optimal solution relying mostly on the root linear relaxation, and in four cases, the optimal solution is reached at the root node. Conversely, while the PI_0-based algorithm continues to find feasible solutions quickly, it struggles to close the optimality gap due to the presence of numerous equivalent solutions as well as less effective pruning due to looser relaxations, causing a significant increase in the number of explored nodes.

Table 3: Runtime % to improve UB and LB for PI_0.

inst./λ	finding optimum	proving optimality
ft06/1.0	0.1%	99.9%
ft06/1.1	6.0%	94.0%
ft06/1.2	7.8%	92.2%
la04/1.0	62.2%	38.8%
la04/1.1	0.2%	-
la04/1.2	0.3%	-

To this end, Table 3 aims to highlight the percentage of time for PI_0 that is spent on 1) finding a feasible solution with optimal objective value, and once it is found, 2) proving its optimality. From the results above, it appears that the majority of runtime is spent on proving optimality. Note that for $\lambda > 1.0$, even though incumbents with optimal objective values are identified early in the execution, the model fails to close the gap.

Valid Inequalities: In Table 4, we show the impact of different families of cuts on the efficiency of PI_0. Symbols T/(%), $\%^r$ and #n have the same meaning as in Table 2. The following PI_0 variants are tested,

- PI_1, statically adding inequalities (9)-(16),
- PI_2, incorporating inequalities (17)-(19),
- PI_{all}, including all the described valid inequalities.

The studied valid inequalities were categorized based on the involvement of processing duration variables, particularly through upper-bound constraints rather than if-then implications. Since processing durations are partitioned across periods with varying costs, minimizing a total weighted sum tightens these upper-bound constraints, especially in low-cost ToU periods. Further, non-overlap is modeled using big-M constraints (8) on variables linked to processing duration variables via inequalities (5), potentially contributing to the weak relaxation in the base model.
The results of Table 4 demonstrate that the second set of inequalities in PI_2 have a stronger impact. Optimality is proven within at most four seconds on all instances, which is a significant reduction in computing time, especially for larger λ values. For the la04 instances, as the time horizon grows, the impact of these inequalities becomes greater as opposed to the trend that PI_0 shows. Conversely, for the ft06 set, both computing time and the number of explored nodes grow as λ increases for all formulations. Further, the second set of inequalities alone reduce the root gap to the same order of magnitude as the much tighter time-indexed model IP2. This is especially evident in the la04 instances: for $\lambda = 1.1$ and $\lambda = 1.2$, the root

Table 4: Comparison of the different proposed PI_0 variants on the ft06 and la04 instances.

	PI_0			PI_1			PI_2			PI_{all}		
inst./λ	T/(%)	$\%^r$	#n	T/(%)	$\%^r$	#n	T/(%)	$\%^r$	#n	T/(%)	$\%^r$	#n
ft06/1.0	0.02	12.4%	1	0.02	12.4%	2	0.01	0.9%	1	0.01	0.9%	1
ft06/1.1	2.21	10.7%	2232	3.07	10.7%	2473	0.40	0.6%	470	0.62	0.6%	386
ft06/1.2	64.40	10.3%	50K	56.92	10.3%	38K	1.96	0.9%	1606	2.45	0.9%	1165
la04/1.0	20.62	12.6%	3910	23.14	12.6%	3700	4.00	0.5%	2237	6.44	0.5%	1900
la04/1.1	(2.8%)	11.8%	612K	(3.3%)	11.8%	276K	0.18	0.0%	75	0.53	0.0%	86
la04/1.2	(5.3%)	11.8%	506K	(5.4%)	11.8%	205K	0.05	0.0%	1	0.14	0.0%	1

gap reduces to 0%, whereas the average number of explored nodes decreases to just 75 and 1, respectively.

On the other hand, while the number of explored nodes is reduced on four of the six sets for PI_1, the computing time remains almost unchanged with respect to PI_0. The performance of the root gap suggests that the inequalities (9)-(16) overburden the linear relaxations at each node without improving the gap, whereas the reduction in the number of nodes could be attributed to bound propagation and locally valid cuts within individual parts of the tree. Further, when combining both families of inequalities in PI_{all}, we notice a slight loss of time performances with respect to PI_2 throughout the whole benchmark set, despite exploring equal or fewer nodes overall. We can conclude that on the tested instances, the PI_2-based B&B is the most-performing algorithm.

Table 5: Root relaxation time and model size comparison.

inst.	IP2	PI_0	PI_2	#rows $\frac{PI_2}{IP2}$
ft06	< 0.1ms	< 0.1ms	< 0.1ms	0.35
la04	214.12s	< 0.1ms	10ms	0.05

Table 5 compares the CPU time required to solve the root node relaxation for IP2, PI_0, and PI_2, along with the ratio of the number of rows between IP2 and PI_2, aggregated by instance. While IP2 provides strong relaxations, its computational cost significantly rises for larger instances. In contrast, the valid inequalities introduced for PI_2 enhance the quality of the root relaxation as shown in Table 4, with negligible impact on the model size and computation time. These results highlight the strength of formulation PI_2 on the tested instances. After adding inequalities (17)-(19), the proposed period-based MILP (a) remains compact, (b) retains its ability to quickly find high-quality feasible solutions, (c) achieves a strong root relaxation, (d) whilst also being able to solve each linear relaxation relatively fast.

5 CONCLUSION AND FUTURE WORK

In this work, we address the job-shop scheduling problem under energy considerations, specifically focusing on minimizing total energy costs within a Time-of-Use (ToU) pricing scheme, denoted as $Jm||TEC$. To achieve this, we propose a novel period-indexed MILP, in which the variables used to compute the cost of the energy usage of operations are indexed based on the periods of the ToU profile. This approach was preferred to a time-indexed alternative to obtain a more compact MILP, as time-indexed formulations can be very effective, but struggle when the considered time span gives rise to large mathematical programs, as it is the case with the reference TI model in the literature.

Our results indicate that the proposed period-indexed model is well-suited for achieving significantly smaller model sizes, and the Branch-and-Bound (B&B) tree node linear relaxations can be computed up to six orders of magnitude faster for some of the tested instances with a larger time span. With instances featuring a time horizon equal to the optimal makespan of the original JSSP instance, the B&B algorithm based on the period-indexed model in its basic setting outperforms the time-indexed one. This is due to the fact that the proposed period-indexed formulation, in spite of weak linear relaxations, allows to find feasible solutions more easily than the time-indexed counterpart, largely due to the model size. However, as the time span is enlarged, the latter benefits of larger scheduling horizons, finding more feasible solutions and then converging more quickly due to the tighter linear relaxation, whereas the former struggles to close the optimality gap.

To address this issue, we explored the addition of different sets of valid inequalities to the proposed period-indexed model in order to tighten its linear relaxation. Three different variants of the base model were tested in which the proposed valid inequalities were alternatively or jointly considered. The results show that inequalities aimed at strengthening non-overlap constraints and tightening the bounds of the period-indexed variables are the most effective. This is because they allow the linear relaxation to be comparable to that of the time-indexed model, despite a smaller number of variables and constraints, hence preserving these advantages while ensuring a much quicker convergence.

Future research directions include strengthening the less-effective valid inequalities and/or adding them dynamically in the B&B tree through tailored separation routines. From a combinatorial perspective, it would certainly be interesting to generalize the piecewise-constant ToU profile and consider more complex cost structures. Another interesting direction would be to consider variable power profiles instead of constant ones. Finally, the flexibility of the period-indexed formulation opens the possibility for its application to other scheduling problems, as well as the incorporation of other constraints, such as release and due dates, as well as setup times, making it a versatile tool for broader industrial applications dealing with energy cost minimization.

REFERENCES

Bley, A. and Linß, A. (2022). Propagation and branching strategies for job shop scheduling minimizing the weighted energy consumption. In *International Conference on Operations Research*, pages 573–580. Springer.

Che, A., Zhang, S., and Wu, X. (2017). Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. *Journal of Cleaner Production*, 156:688–697.

Cheng, J., Chu, F., Chu, C., and Xia, W. (2016). Bi-objective optimization of single-machine batch scheduling under time-of-use electricity prices. *RAIRO - Operations Research*, 50:715–732.

Conforti, M., Cornuéjols, G., Zambelli, G., Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer programming models*. Springer.

Dauzère-Pérès, S., Ding, J., Shen, L., and Tamssaouet, K. (2024). The flexible job shop scheduling problem: A review. *European Journal of Operational Research*, 314(2):409–432.

Ding, J. Y., Song, S., Zhang, R., Chiong, R., and Wu, C. (2016). Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Transactions on Automation Science and Engineering*, 13:1138–1154.

Fisher, H. and Thompson, G. (1963). Probabilistic learning combinations of local job-shop scheduling rules. *Prentice Hall, Englewood Cliffs, New Jersey*, pages 225–251.

Gaggero, M., Paolucci, M., and Ronco, R. (2023). Exact and heuristic solution approaches for energy-efficient identical parallel machine scheduling with time-of-use costs. *European Journal of Operational Research*, 311:845–866.

Garey, M. R., Johnson, D. S., and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117–129.

Graham, R., Lawler, E., Lenstra, J., and Kan, A. (1977). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326.

Ho, M. H., Hnaien, F., and Dugardin, F. (2022). Exact method to optimize the total electricity cost in two-machine permutation flow shop scheduling problem under time-of-use tariff. *Computers and Operations Research*, 144.

International Energy Agency (2023). World energy outlook 2023. https://www.iea.org/reports/world-energy-outlook-2023. Accessed: 2024-10-01.

Jiang, E. and Wang, L. (2020). Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity prices. *Knowledge-Based Systems*, 204.

Lawrence, S. (1984). Resouce constrained project scheduling: An experimental investigation of heuristic scheduling techniques (supplement). *Graduate School of Industrial Administration, Carnegie-Mellon University*.

Masmoudi, O., Delorme, X., and Gianessi, P. (2019). Job-shop scheduling problem with energy consideration. *International Journal of Production Economics*, 216:12–22.

Park, M. J. and Ham, A. (2022). Energy-aware flexible job shop scheduling under time-of-use pricing. *International Journal of Production Economics*, 248.

Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems*. Springer, 5th edition.

Tian, Z. and Zheng, L. (2024). Single machine parallel-batch scheduling under time-of-use electricity prices: New formulations and optimisation approaches. *European Journal of Operational Research*, 312:512–524.