European Journal of Operational Research 308 (2023) 1091–1109

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Job scheduling under Time-of-Use energy tariffs for sustainable manufacturing: a survey

Daniele Catanzaro[a], Raffaele Pesenti[b], Roberto Ronco[c,*]

[a] Center for Operations Research and Econometrics, Université Catholique de Louvain, Voie du Roman Pays 34, Louvain-la-Neuve 1348, Belgium
[b] Department of Management, University Ca' Foscari, San Giobbe, Cannaregio 837, Venice I-30121, Italy
[c] Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Viale Causa 13, Genova 16145, Italy

A R T I C L E I N F O

Article history:
Received 8 January 2022
Accepted 14 January 2023
Available online 20 January 2023

Keywords:
Scheduling
Energy efficiency
Time-of-use tariffs
Sustainable manufacturing
Combinatorial optimization

A B S T R A C T

The combined increase of energy demand and environmental pollution on a global scale is forcing a re-thinking, in sustainable terms, of energy supply policies and production models. To flatten demand peaks in power plants, energy suppliers adopted pricing policies that stimulate a change in the consumption practices of customers. One example of such policies is the Time-of-Use (TOU)-based tariffs, which encourage electricity usage at off-peak hours using low prices while penalizing peak hours with higher prices. To avoid a sharp rise in energy supply costs, the manufacturing industry must carefully reschedule the production processes, by shifting them toward less expensive periods. TOU-based tariffs impose specific constraints on the completion of the jobs involved in the production processes as well as a partitioning of the time horizon of the production into a set of time slots, whose associated cost becomes part of the optimization objective. In this article, we review the flourishing literature on job scheduling under TOU-based energy tariffs. Our purpose is to provide researchers and practitioners with a framework that may guide them toward the most important theoretical results on the topic as well as the most prominent practical applications in sustainable manufacturing.

1. Introduction

The rise of worldwide energy consumption in the last decades, together with the consequent environmental pollution, has become a compelling matter for global demand-supply. The World Energy Outlook 2019 reported 14,314 million tonnes of oil-equivalent demand from primary energy consumption, corresponding to 33.2 gigatonnes of CO_2 emission in the sole 2018 (International Energy Agency, 2019). The sectors related to manufacturing, agriculture, mining, and construction collectively consumed 54% of the worldwide energy demand in 2012 (International Energy Agency, 2016). In 2018, the total energy consumption of the sole manufacturing sector amounted to 19.436 trillion British thermal units (Energy Information Administration (EIA), 2021), and the above sectors altogether were forecast to grow at a rate of 1.2% per year until 2040 (International Energy Agency, 2016). Rethinking the production processes under a sustainable lens, and simultaneously fostering environment-aware consumption practices in customers, appears to be a necessary condition to invert this trend. One of the first actions undertaken by energy suppliers consisted in trying to flatten the demand peaks of power plants by devising strategies aimed at reducing the high economic and environmental burdens related to the generation of high energy loads in short periods of time. These strategies mostly consist of pricing policies that stimulate a change in the consumption practices of customers (Hu et al., 2018). One example of such policies is provided by the *Time-of-Use* (TOU)-based tariffs, which spur electricity usage at off-peak hours through low prices while penalizing peak hours with higher prices. The identification of the optimal prices taking into account the inherent stochasticity of the demand can be carried out, e.g., by using robust optimization approaches, such as those described in Hu et al. (2018), or stochastic programming models, such as those discussed in Nikzad & Samimi (2020). Overall, TOU-based tariff policies proved successful so far to flatten the peaks of demand, by ensuring, at the same time, good service stability (Chawla et al., 2017; Hu et al., 2018). How long these policies will prove effective in containing a world-scale increase in energy consumption, however, still remains hard to foresee. It is plausible that energy suppliers will rely more and more upon such policies to face possible energy shortages or price fluctuations caused by international instabilities such as the current Russian-Ukrainian conflict

* Corresponding author.
E-mail addresses: daniele.catanzaro@uclouvain.be (D. Catanzaro), pesenti@unive.it (R. Pesenti), roberto.ronco@edu.unige.it (R. Ronco).

https://doi.org/10.1016/j.ejor.2023.01.029
0377-2217/

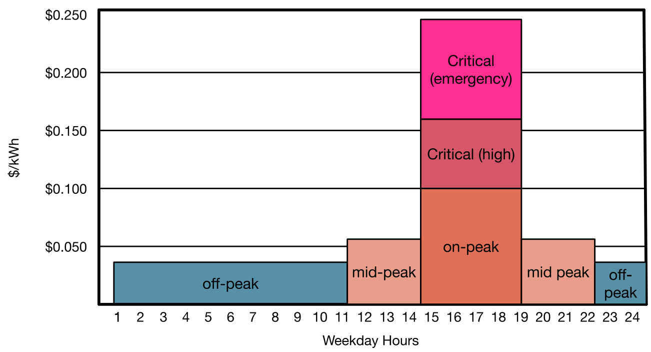

Fig. 1. An example of a TOU pricing scheme from Chen & Zhang (2019). The figure shows a possible hourly variation of TOU costs, expressed in dollars per kilowatt-hour ([$/KWh]), by highlighting the off-peak, mid-peak, and on-peak periods of the day. An increase in energy demand results in higher TOU costs, which account for the considerable costs of handling the generation of high energy generation and, at the same time, discourage further customers demand.

or Indo-Pacific tensions (Pereira et al., 2022). On the other hand, the COVID-19 pandemic has dramatically changed the face of the global supply chain, prompting firms and enterprises to reshore in the last few years (Karatzas et al., 2022). Energy-intensive industries already have to handle the effect of increasing restrictions on carbon emissions as well as the geopolitical aftermath of the Russian-Ukrainian conflict on energy provisioning (Lambert et al., 2022). The consequent changes in the related business models will stress the capabilities of the energy pricing schemes to adapt to the novel demand-supply dynamics in the near future.

In the context of manufacturing, a natural reaction to TOU-based tariffs consists in rescheduling the production processes during periods characterized by low energy supply costs. Job scheduling under TOU-based tariffs (or *TOU scheduling* for short) is similar to classical job scheduling problems: the processing order of a given set of jobs must be identified and carried out on one or more machines (Pinedo, 2016). Moreover, the processing of the jobs must comply with time requirements (e.g., a due date or a common deadline Pinedo, 2016). However, the presence of TOU-based tariffs induces an implicit partitioning of the time horizon of the production into a set of time slots, each associated with a non-negative cost called *TOU costs* (Chen & Zhang, 2019) or *TOU prices* (Ding et al., 2016). This partitioning is usually referred to as the *TOU pricing scheme* (Che et al., 2017; Ding et al., 2016; Soares et al., 2021), whereas the sum of the TOU costs of the time slots used for processing in the considered time horizon is referred to as the *Total Energy Cost* (TEC). Generally, the TEC is the objective of a TOU scheduling problem. In the context of multi-objective optimization problems, the TEC may add to other classical objectives in scheduling, such as the *makespan* or the *total weighted tardiness* (Pinedo, 2016). Figure 1 shows an example of a typical hourly variation of TOU costs in response to the off-peak, mid-peak, and on-peak periods of energy demand throughout the day.

The literature on sustainable manufacturing offers several recent surveys on energy-efficient scheduling, each one providing a specific perspective on the topic. For instance, Gao et al. (2020) and Terbrack et al. (2021) independently focused on production planning for intelligent systems, by proposing a general classification of the most important problems in sustainable manufacturing and the corresponding solution approaches. Similarly, Gahm et al. (2016) studied the impact of energy-aware practices in sustainable production, by proposing a classification of energy-efficient job scheduling based on energetic coverage, energy supply, and energy demand. Para et al. (2022) delved into energy-efficient job shop scheduling problems by developing a comprehensive survey specifically focused on metaheuristics as solution approaches. Alvarez-Meaza et al. (2021) also considered a vast number of research articles on green scheduling, with the intent of outlining the most promising and compelling research directions for the scientific community. Choudhury et al. (2020) focused on the technological aspects of manufacturing, by analyzing the service supply chain and its environmental-related concerns. Despeisse et al. (2012) proposed a conceptual framework to model single factory units as ecosystems aimed at enabling sustainable performance improvements while limiting environmental pollution and unrestrained exploitation of natural resources. Giret et al. (2015) reviewed energy-efficiency operations scheduling by exploring the conditions for its sustainability. The authors distinguished between approaches based on the input data (e.g., as machines, jobs, and scheduling horizon), approaches based on the environmental output (e.g., pollution, waste), and approaches based on mixed objectives. Renna & Materi (2021) also analyzed energy-efficient manufacturing systems from the point of view of sustainability. The authors intended to review the integration of renewable sources in manufacturing systems to promote further research in this direction. Cai et al. (2022) pursued the same goal, but they tackled the discussion of the topic by specifically focusing on four technological processes that enable sustainable manufacturing: energy monitoring, evaluation, optimization, and benchmarking. Finally, Yusuf et al. (2021) proposed a review of energy minimization strategies for tools in manufacturing processes. None of the above surveys, however, comprehensively discusses the models, methods, and algorithms for job scheduling under TOU-based energy tariffs presented in the literature. In this article, we close this gap, by providing researchers and practitioners with (i) a framework that summarizes the most important theoretical results and practical applications on the topic, and (ii) a guide that may direct new research efforts toward unexplored directions in TOU scheduling for sustainable manufacturing. In Section 2, we will introduce some notation and definitions aimed at both stating the general TOU scheduling paradigm and describing the characterizing features of the different versions of the problem that occur in the literature. We will present a possible classification of these versions in Section 3, by discussing the theoretical and technological motivations at their core as well as the respective approaches proposed to solve them. We will devote Section 4 to a subset of specific versions of TOU scheduling that cannot be framed in a standard taxonomy, and that may either guide toward novel research branches or be potentially inspiring in the context of industrial applications. In Section 5, we will draw some conclusions, by highlighting the limits of current approaches and providing future research directions.

2. Notation and problem statement

In this section, we formalize the job scheduling problem in presence of TOU-based energy tariffs. To this end, we first need to introduce some notation and definitions that will prove useful in the remainder of the article.

Given two positive integers N and K, we denote $\mathcal{J} = \{1, 2, \ldots, N\}$ and $\mathcal{T} = \{1, 2, \ldots, K\}$ as a set of N jobs to be processed on a single-job processing machine and a set of K consecutive time slots constituting the time horizon of the production, respectively. We denote p_j as a positive integer encoding the *processing time* of the job $j \in \mathcal{J}$, i.e., the number of time slots required to process j on the given machine. Similarly, we denote $c_t \in \mathbb{N}$ as the TOU cost associated with the time slot $t \in \mathcal{T}$. For the sake of conciseness, we denote $\{c_t\}$ as the set of the time slots costs c_t, for $t \in \mathcal{T}$.

As in Fang et al. (2016), we say that $\{c_t\}$ is *pyramidal* if there is some $t' \in \mathcal{T}$ such that $c_1 < c_2 < \ldots < c_{t'-1} < c_{t'} > c_{t'+1} > \ldots > c_{K-1} > c_K$. Moreover, we say that $\{c_t\}$ is *non-decreasing* (respectively, *non-increasing*) if $c_t \leq c_{t'}$ (respectively, $c_t \geq c_{t'}$) for all $t, t' \in \mathcal{T}$, $t < t'$.

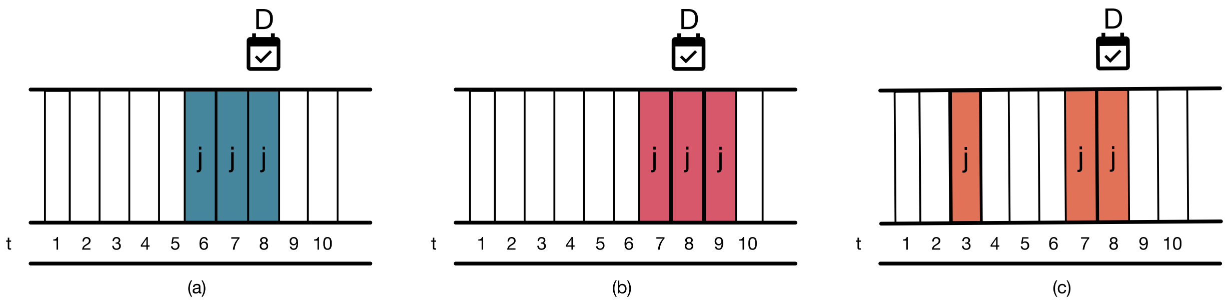

Fig. 2. An example of three possible schedules for a dummy instance of the BJSTP involving a single job j with processing time 3, a time horizon consisting of 10 time slots, a deadline $D = 8$, and TOU costs all equal to one and omitted for the sake of clarity. The schedule shown in Fig. 2(a) is non-preemptive while the one shown in Fig. 2(c) is preemptive. The schedule shown in Fig. 2(a) is feasible as the processing of the job is completed within the deadline. The other two schedules are unfeasible. In particular, the schedule shown in Fig. 2(b) violates the deadline and the schedule shown in Fig. 2(c) violates non-preemption.

We define a *job-time slot assignment* as a pair $(j, \mathcal{T}_j)$ such that j is a job in $\mathcal{J}$, and $\mathcal{T}_j$ is a non-empty subset of p_j time slots in $\mathcal{T}$ during which j is processed. We define a *schedule* S as a set of job-time slot assignments satisfying the following two conditions: (i) exactly one pair $(j, \mathcal{T}_j)$ exists for each $j \in \mathcal{J}$ and (ii) for any pair of distinct jobs $j', j'' \in \mathcal{J}$, the intersection $\mathcal{T}_{j'} \cap \mathcal{T}_{j''}$ is empty, i.e.,

$$S = \{(j, \mathcal{T}_j) : \emptyset \neq \mathcal{T}_j \subseteq \mathcal{T}, \forall\, j \in \mathcal{J}, \text{ and } \mathcal{T}_{j'} \cap \mathcal{T}_{j''} = \emptyset \;\forall\; j', j'' \in \mathcal{J}, j' \neq j''\}. \quad (1)$$

A schedule S is *preemptive* if, for some job $j \in \mathcal{J}$, $\mathcal{T}_j$ contains p_j non-consecutive slots, and *non-preemptive* otherwise. For example, Fig. 2 shows three possible schedules for a job j characterized by a processing time $p_j = 3$ over a time horizon of 5 slots. The schedule shown in Fig. 2(a) is non-preemptive as job j is processed consecutively in 3 time slots. The schedule shown in Fig. 2(c) is instead preemptive, as job j is not processed consecutively in a number of time slots equal to p_j.

We denote $\mathcal{S}$ as the set of the possible schedules of the jobs in $\mathcal{J}$ over the time horizon $\mathcal{T}$, and we define the *Total Energy Cost* (TEC) of a schedule $S \in \mathcal{S}$ as

$$E(S) = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}_j} c_t, \quad (2)$$

i.e., as the sum of the TOU costs associated with the job-time slot assignments in S. Similarly, we refer to the overall amount of time necessary to process the jobs in $\mathcal{J}$ as the *makespan* $C_{\max}$ of S, and we define it as

$$C_{\max}(S) = \max_{t \in \bigcup_{j \in \mathcal{J}} \mathcal{T}_j} t. \quad (3)$$

Moreover, the deadline of a schedule S is a positive integer D that represents the time by which all jobs must be completed. Hence, the makespan of S must not exceed D, i.e., $C_{\max}(S) \leq D$. Then, given the above notation and definitions, the basic version of the job scheduling problem in presence of TOU-based tariffs reads as follows:

The Basic Job Scheduling with TOU tariffs Problem (BJSTP)**.** Given a set of jobs $\mathcal{J}$, a set of processing times $\{p_j\}$ associated with the jobs in $\mathcal{J}$, a time horizon $\mathcal{T}$, a set of TOU costs $\{c_t\}$ associated with the time slots in $\mathcal{T}$, and a deadline D, find a non-preemptive schedule $S \in \mathcal{S}$ that minimizes the total energy cost (2) and whose makespan (3) is less than or equal to the given deadline, i.e.,

$$\min_{S \in \mathcal{S}} \quad E(S) \quad (4)$$

$$s.t. \quad C_{\max}(S) \leq D. \quad (5)$$

For example, consider again the three alternative schedules shown in Fig. 2. Then, the schedule shown in Fig. 2(a) encodes a

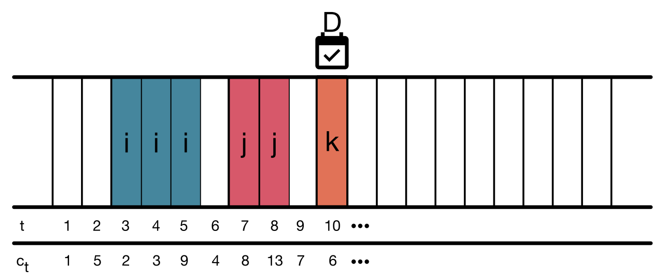

Fig. 3. An example of a possible schedule for an instance of the BJSTP having three jobs $\{i, j, k\}$, processing times $\{3, 2, 1\}$, a time horizon of 10 time slots, and TOU costs $c_t = \{1, 5, 2, 3, 9, 4, 8, 13, 7, 6\}$. The deadline D here is assumed to be equal to 10.

feasible solution to the BJSTP. In contrast, the schedule shown in Fig. 2(b) is infeasible, as part of job j is processed after the deadline. Similarly, the schedule shown in Fig. 2(c) is infeasible as well because it is preemptive, even if it satisfies the deadline.

Figure 3 shows an example of a possible schedule for an instance of the BJSTP having three jobs $\{i, j, k\}$ with processing times $\{3, 2, 1\}$, a time horizon of 10 time slots, TOU costs $c_t = \{1, 5, 2, 3, 9, 4, 8, 13, 7, 6\}$, and deadline $D = 10$. This schedule has a TEC equal to 41 and a makespan of 10. Observe that if job i, j, and k were moved to slots $\{2, 3, 4\}$, $\{6, 7\}$ and $\{1\}$, respectively, the resulting schedule would achieve a makespan equal to 7 and a TEC equal to 23, which is provably minimum.

We conclude this section by observing that the literature on the BJSTP often assumes an identical power consumption profile for all jobs whenever a specific job-per-job profile is unavailable. The time slot costs, therefore, are often expressed in terms of currencies (e.g., dollars Anghinolfi et al., 2021; Chen et al., 2021; Wang et al., 2018a). When the job consumption profiles are instead available, the time slot costs are expressed as currency over power (e.g., dollar over Kilowatt-hour), as in Fig. 1 (see also Aghelinejad et al., 2018a; Fang et al., 2016). In such a case, the considered time horizon $\mathcal{T}$ spans over a day, and a time slot $t \in \mathcal{T}$ consists of an hour.

3. A classification of the literature on the BJSTP

In this section, we present a possible taxonomy of the different versions of the BJSTP that occur in the literature on TOU scheduling. Figures 4 and 5 provide a graphical representation of this taxonomy and may assist the reader in understanding the relationships and the conceptual dependencies among these versions.

Throughout the article, we will make use of Graham et al. (1977)'s *three-field* notation, or $\alpha|\beta|\gamma$ notation for short, which is commonly used in job scheduling to describe the problems at hand in a compact and easy-to-read form. The α field character-

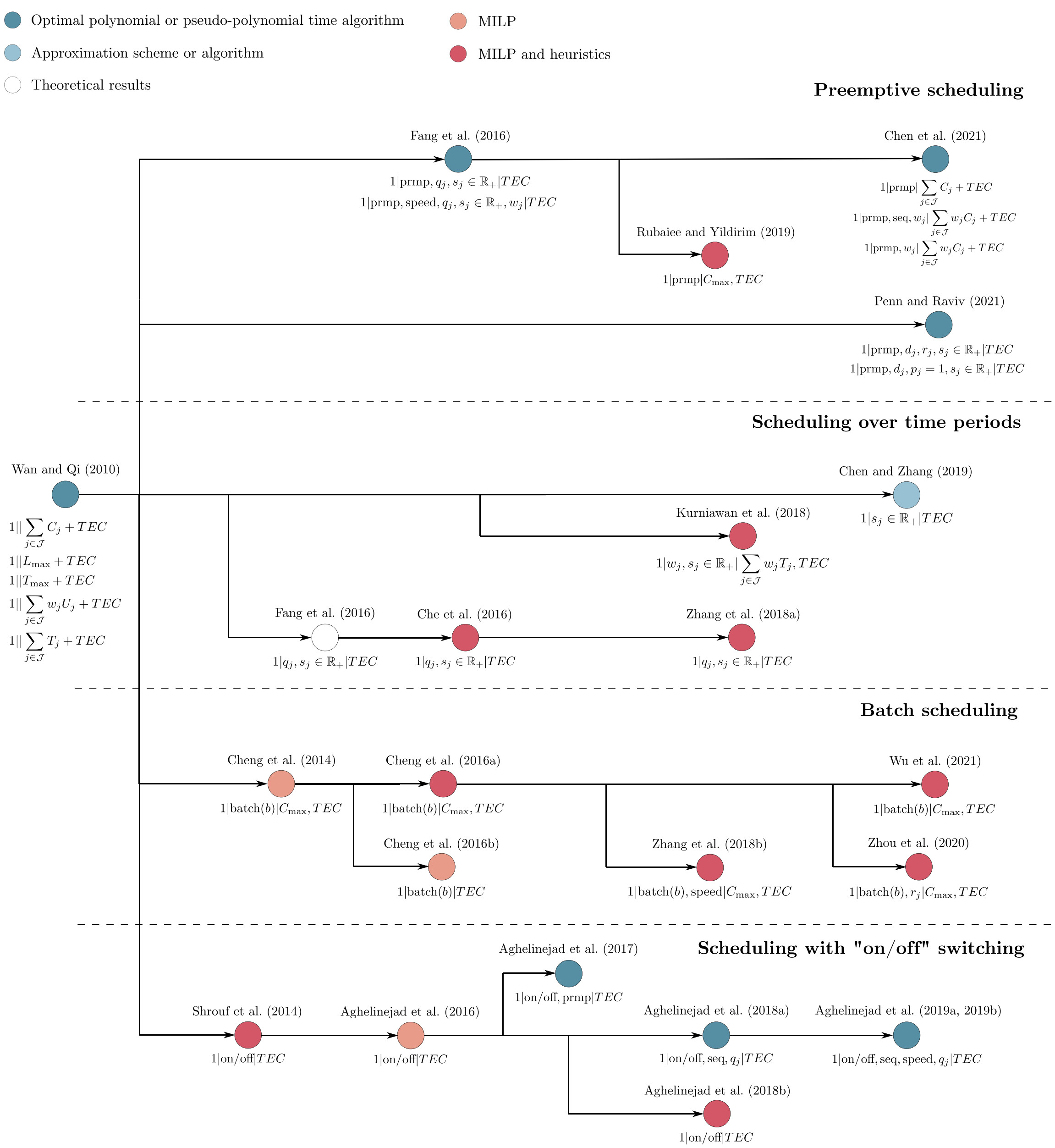

Fig. 4. A graphical representation of the development of the literature on single-machine versions of the BJSTP. Each node, displayed as a circle, corresponds to a specific work in the literature, while the directed edges, depicted as pointed arrows, connect different works that are directly related both historically and content-wise. Furthermore, the color of each node depends on the solution approach developed by the authors to solve the related problem, according to the rules reported in the top-left corner of the image.

izes the processing environment. For example, α takes value 1 in the case of a single machine, whereas α is denoted as Pm in the case of m parallel and identical machines (where P stands for parallel and identical, and m accounts for the number of considered machines). The β field specifies the constraints and the processing restrictions, such as jobs preemption or release dates, hereafter denoted as prmp and r_j, respectively. Finally, the γ field refers to the optimization objectives, such as the minimization of the makespan or the TEC. For instance, the BJSTP can be represented by employing the three-field notation as $1||TEC$, since it only involves a single machine, and requires finding a non-preemptive schedule that minimizes the TEC. We refer the reader interested in a more in-depth discussion on the three-field notation to Pinedo (2016)'s classical book. Here, we just summarize the possible values that the three fields may take in Table 1. We also discuss the development of the BJSTP for single and multiple-machine environments separately, because most of the properties for single-machine problems are very hard to generalize to the case of multiple machines. In fact, only a few results in single-machine versions of the BJSTP directly translate into insights and results for the multiple-machine versions. Finally, when referring to algorithms, we will use the acronyms listed in Table 2, and we will assume (unless specified otherwise) that the machines are single-job processing.

3.1. Single-machine versions of the BJSTP

The earliest work dealing with a single-machine version of the BJSTP has been proposed by Wan & Qi (2010). The authors considered several versions of the BJSTP differing from one another for the type of objective function considered, including (i) the *total flow time* (i.e., the $1||\sum_{j\in\mathcal{J}} C_j + TEC$ under Graham et al. (1977)'s three-field notation); (ii) the *maximum lateness* (i.e., the

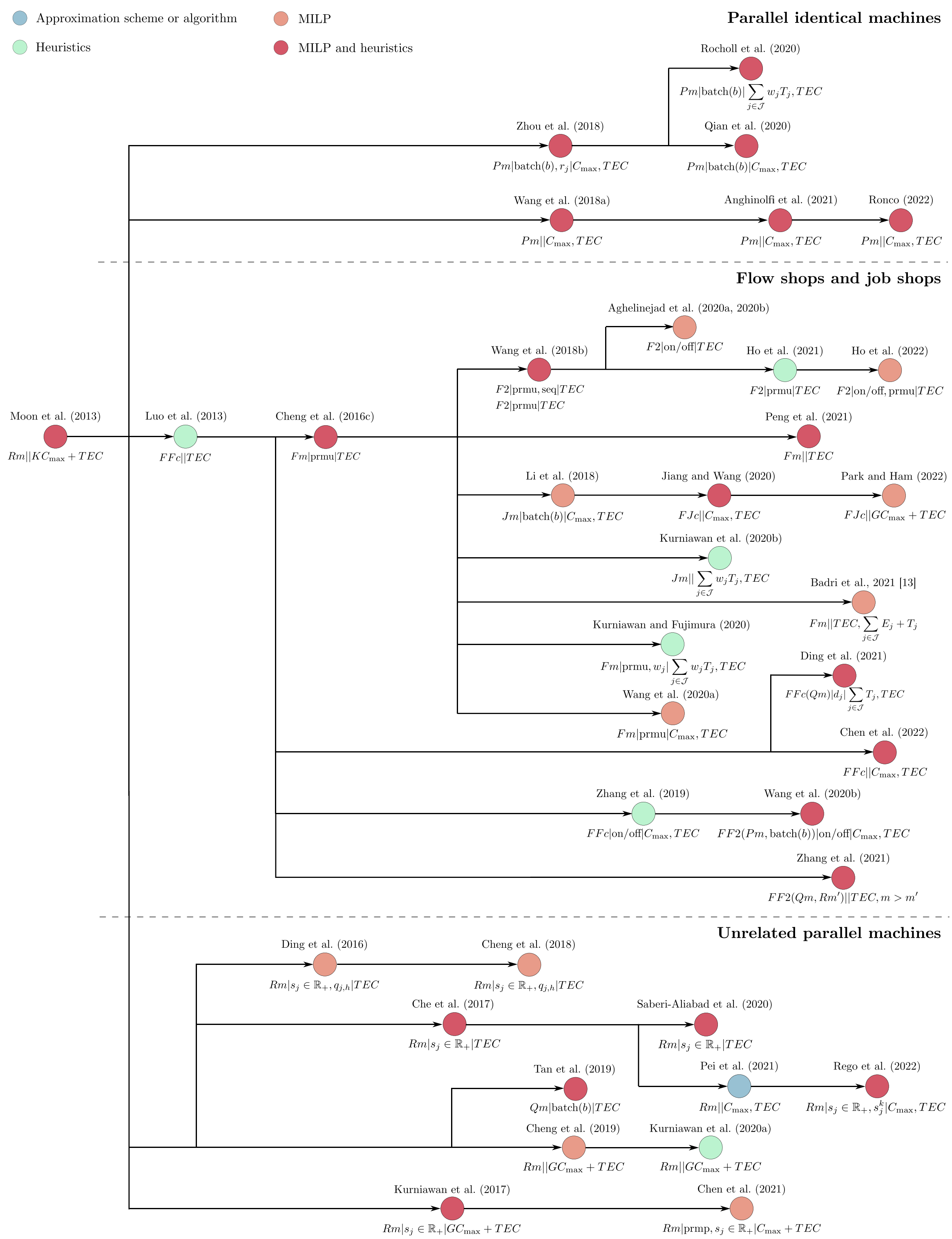

Fig. 5. A graphical representation of the development of the literature on the versions of the BJSTP with multiple machines. Each node, displayed as a circle, corresponds to a specific work in the literature, while the directed edges, depicted as pointed arrows, connect different works that are directly related both historically and content-wise. Furthermore, the color of each node depends on the solution approach developed by the authors to solve the related problem, according to the rules reported in the top-left corner of the image.

$1||L_{\max} + TEC$); (iii) the *maximum tardiness* (i.e., the $1||T_{\max} + TEC$); (iv) the *weighted number of tardy jobs* (i.e., the $1||\sum_{j\in\mathcal{J}} w_jU_j + TEC$); and (v) the *total tardiness* (i.e., the $1||\sum_{j\in\mathcal{J}} T_j + TEC$). The authors proved the strong $\mathcal{NP}$-hardness of all of these versions by using a reduction from the *3-Partition Problem* (Garey & Johnson, 1990). This reduction technique, which often recurs in the literature, has inspired complexity results for other versions of the BJSTP discussed later in this section, such as those considered by Fang et al. (2016) and Chen & Zhang (2019).

Besides showing general intractability results, Wan & Qi (2010) also showed that the above versions of the BJSTP can be solved in polynomial time if specific assumptions hold on prob-

Table 1
Graham et al. (1977)'s three-field notation for classical scheduling problems.

Field	Characteristic	Meaning
α field	1	single machine
	Pm	m parallel identical machines
	Qm	m parallel machines with different speeds
	Rm	m unrelated parallel machines)
	Fm	flow shop on m machines
	Jm	job shop on m machines
	FFc	flexible flow shop on c machines
	FJc	flexible job shop on c machines
β field	r_j	release dates
	d_j	due dates
	w_j	weights
	prmp	preemption
	batch(b)	parallel batch processing
	prmu	permutation
	q_j	jobs power demands
	$q_{j,h}$	machine-dependent jobs power demands
	seq	fixed jobs processing sequence (on a single machine)
	$s_j \in \mathbb{R}_+$	non-negative real start times
	s_j^k	sequence-dependent setup times
	on/off	machines regulation via "on/off" switching mechanism
γ field	$C_{\max}$	makespan
	$L_{\max}$	maximum lateness
	$T_{\max}$	maximum tardiness
	$\sum w_j C_j$	total weighted completion time
	$\sum w_j T_j$	total weighted tardiness
	$\sum w_j U_j$	weighted number of tardy jobs

Table 2
Summary of the acronyms for heuristics and metaheuristics used in the literature on the BJSTP.

Algorithm	Acronym
Ant Colony Optimization	ACO
Evolution Algorithm	EA
Genetic Algorithm	GA
Iterated Local Search	ILS
Local Search	LS
Multi-Population Genetic Algorithm	MPGA
Non-dominated Sorted Genetic Algorithm	NSGA
Particle Swarm Optimization	PSO
Strength Pareto-archived Evolutionary Algorithm	SPEA
Single-Population Genetic Algorithm	SPGA
Tabu Search	TS
Variable Neighborhood Search	VNS

lem data. For example, in the case of the $1||\sum_{j\in\mathcal{J}} C_j + TEC$, the authors observed that if the time slot costs $\{c_t\}$ are non-increasing over time, then there exists at least an optimal solution (schedule) to the problem such that the set of jobs $\mathcal{J}$ obeys to the so-called *Shortest Processing Time* (SPT) first order, i.e., a schedule in which the jobs are sorted such that the processing time of job $j \in \mathcal{J}$ precedes the one of job $j' \in \mathcal{J} \setminus \{j\}$ whenever $p_j < p_{j'}$. Wan & Qi (2010) showed that this optimal solution can be computed in $O(K^2)$ via dynamic programming. The main idea at the core of the algorithm can be summarized as follows: first, sort the jobs according to the SPT rule; subsequently, given the first n jobs in the sorted job sequence, $n \leq N$, denote by $f_{n,k}$ the total cost of the schedule obtained when the completion time of the nth job is no greater than k. Observe that the nth job can be completed either at time slot k or before it. Hence, the optimal schedule of the sorted jobs can be obtained by recursively computing the following expression

$$f_{n,k} = \min\left\{k + \sum_{i=k-p_n+1}^{k} c_i + f_{n-1,k-p_n},\ f_{n,k-1}\right\} \quad \forall n \in \mathcal{J} \setminus \{1\},\ k \in \mathcal{T}, \tag{6}$$

where the first argument of the min$\{\cdot\}$ function is the cost paid when the nth job ends at the time slot k, while the second argument is the cost paid when nth job ends before the time slot k. The initial condition for (6) is given by

$$f_{1,k} = \min_{p_1 \leq l \leq k}\left\{l + \sum_{i=l-p_1+1}^{l} c_i\right\}, \quad k = p_1, p_1+1, \ldots, K, \tag{7}$$

where p_1 is the processing time of the first job in the sequence, i.e., the shortest processing time, and $f_{1,k} = \infty$ for $k = 1, 2, \ldots, p_1 - 1$. The optimal cost is given by (6) when $j = N$ and $k = K$. Because there are NK states for the dynamic program and evaluating (6) takes a constant time, computing $f_{N,K}$ requires $O(NK)$ time. Since evaluating (7) for each $k = p_1, p_1+1, \ldots, K$ requires $O(K^2)$ time, the running time of the algorithm is $O(NK + K^2) \sim O(K^2)$. The authors showed that this approach can be appropriately adapted to solve not only the other versions of the BJSTP (see Table 3), but also a more general one characterized by a *convex* set of costs $\{c_t\}$, i.e., costs that are contained in the image of some convex non-increasing function. As for the complexity results, Wan & Qi's dynamic programming approach inspired several solution approaches for subsequent single-machine versions of the BJSTP, such as those described in Chen & Zhang (2019); Chen et al. (2021); Fang et al. (2016); Penn & Raviv (2021).

The earliest versions of the BJSTP considered *homogeneous* time slots, i.e., time slots characterized by the same duration (see, e.g., Fig. 6(a)). The literature on the BJSTP, however, also includes versions of the problem characterized by *heterogeneous* time slots Wan & Qi (2010). In these versions, the time horizon – usually encoded as an interval $[0, K] \in \mathbb{R}$ – is partitioned into subintervals with possibly non-integral duration, hereinafter referred to as *time periods* (see, e.g., Fig. 6(b) and (c)). The characteristics of the time periods may affect schedules in a non-trivial way. For example, by referring to Fig. 6(a), the processing times of the jobs can be either integer or fractional. In the case of integer processing times, the costs associated with the processing of the jobs are the usual $\{c_t\}$; in the case of fractional processing times, instead, these costs are fractions of $\{c_t\}$. The situation becomes more complex in the cases of Fig. 6(b) and (c). In these cases, the jobs have a

Table 3
Optimal polynomial and pseudo-polynomial time algorithms for different versions of the BJSTP.

Article	Problem	Computational complexity of the solution algorithm	Assumptions
Wan & Qi (2010)	$1\|\|\sum_{j\in\mathcal{J}} C_j + TEC$	$O(K^2)$	$\{c_k\}$ non-increasing
		$O(N(N^2+\log\{K\}))$	$\{c_k\}$ convex non-increasing
	$1\|\|L_{\max}+TEC$	$O(K^2)$	$\{c_k\}$ non-increasing
		$O(K\log K)$	$\{c_k\}$ convex non-increasing
	$1\|\|T_{\max}+TEC$	$O(K^2)$	$\{c_k\}$ non-increasing
		$O(K\log K)$	$\{c_k\}$ convex non-increasing
	$1\|\|\sum_{j\in\mathcal{J}} w_jU_j + TEC$	(a) $O(K^2)$ or (b) $O(NK^3)$	Tardy jobs as (a) lost sales or (b) backlogging
	$1\|\|\sum_{j\in\mathcal{J}} T_j + TEC$	$O(N^4K^3)$	$\{c_k\}$ monotone non-increasing
Fang et al. (2016)	$1\|\text{prmp}, s_j\in\mathbb{R}_+, q_j\|TEC$	No explicit expression.	
	$1\|s_j\in\mathbb{R}_+, q_j\|TEC$	No explicit expression.	$\{c_k\}$ pyramidal
	$1\|\text{prmp, speed}, s_j\in\mathbb{R}_+, q_j, w_j\|TEC$	No explicit expression.	
Chen & Zhang (2019)	$1\|s_j\in\mathbb{R}_+\|TEC$	$O(\|\hat{\mathcal{T}}\|)$	The set of time periods $\hat{\mathcal{T}}$ has 1 valley
		$O(\|\hat{\mathcal{T}}\|^2N^{2r-1})$	$\hat{\mathcal{T}}$ has 2 valleys, $\|\{p_j, j\in\mathcal{J}\}\| = r$
		$O((\sum_{j\in\mathcal{J}} p_j)^l(N+\|\hat{\mathcal{T}}\|^l))$	$\hat{\mathcal{T}}$ has $l>2$ valley
		$O(\|\hat{\mathcal{T}}\|(\|\hat{\mathcal{T}}\|+N)N^3)$	Bounded lateness, $\hat{\mathcal{T}}$ has 1 valley
		$O(\|\hat{\mathcal{T}}\|^{\|\hat{\mathcal{T}}\|+2}N^{3\|\hat{\mathcal{T}}\|-1})$	Bounded total completion times; $\hat{\mathcal{T}}$ has 1 valley
Chen et al. (2021)	$1\|\text{prmp}\|\sum_{j\in\mathcal{J}} C_j + TEC$	No explicit expression.	
	$1\|\text{prmp, seq}, w_j\|\sum_{j\in\mathcal{J}} w_jC_j + TEC$	$O(NK)$	
Penn & Raviv (2021)	$1\|\text{prmp}, s_j\in\mathbb{R}_+, d_j, r_j\|TEC$	$O(N^2(N+K)(N\log_2 N+\log_2 N+K))$	
	$1\|\text{prmp}, s_j\in\mathbb{R}_+, d_j, p_j=1\|TEC$	$O(NK^2)$	
Aghelinejad et al. (2017)	$1\|\text{on/off, prmp}\|TEC$	$O(K^3)$	
Aghelinejad et al. (2019a,b)	$1\|\text{on/off, seq}\|TEC$	$O(K^3)$	
Aghelinejad et al. (2018a)	$1\|\text{on/off, seq}, q_j\|TEC$	$O(K^3)$	
	$1\|\text{on/off, speed, seq}, q_j\|TEC$	$O(K^2V(K+V))$, $V\in\mathbb{Z}_+$	
Wang et al. (2018b)	$F2\|\text{prmu, seq}\|TEC$	$O(NK^4)$	

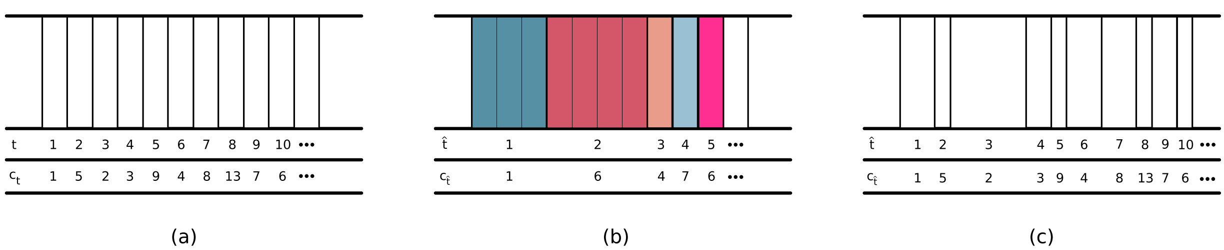

Fig. 6. Different possible ways of partitioning the time horizon. Figure 6(a) shows a partitioning of the time horizon into a set of 10 (homogeneous) time slots. Figure 6(b) shows a partitioning of the time horizon into a set of 5 time periods with integer lengths, namely 3, 4, 1, 1, and 1, with TOU cost 1, 6, 4, 7, and 6, respectively. Finally, Fig. 6(c) shows a partitioning of the time horizon into a set of 10 heterogeneous time periods having possibly non-integral duration.

non-negative real start time and the TEC becomes the integral of a piecewise-constant map that generally associates each time period $[a_i, a_{i+1}] \in [0, K]$, $i \in \{1, \dots, L-1\}$ in a given set $\hat{\mathcal{T}}$ of $L > 0$ time periods to a TOU cost. Both situations, described e.g., in Fang et al. (2016) and Chen & Zhang (2019), possibly constitute close-to-real scenarios for production scheduling systems.

In the next sections, we describe versions of the BJSTP that involve a single machine, according to the characteristics of the jobs and the processing environment. Specifically, in Section 3.1.1 we present versions of the BJSTP that allow jobs to be preempted. In Section 3.1.2 we present versions of the BJSTP that require non-preemptive scheduling over a time horizon partitioned into time periods. In Section 3.1.3 we present versions of the BJSTP that involve batch-processing machines. Finally, in Section 3.1.4 we present versions of the BJSTP that consider machines with power transients between different operational states.

3.1.1. Preemptive versions of the BJSTP

Penn & Raviv (2021) considered the $1|\text{prmp}, s_j \in \mathbb{R}_+, d_j, r_j|TEC$, i.e., the problem of minimizing the TEC of a preemptive schedule for a set of jobs with release and due dates over a time horizon partitioned into a set of time periods, as in Fig. 6(c). The authors proved that this scheduling problem reduces to a special case of the min-cost flow problem on a directed bipartite graph (see Derigs, 1988). This result allowed Penn and Raviv to adapt Brenner's algorithm (Brenner, 2008) to the $1|\text{prmp}, s_j \in \mathbb{R}_+, d_j, r_j|TEC$, by ultimately enabling its exact resolution in $O(N^2(N+K)(N\log_2 N + \log_2 N + K))$. Penn and Raviv also considered a special version of the problem characterized by having no release dates and processing times equal to 1, i.e., the $1|\text{prmp}, s_j \in \mathbb{R}_+, d_j, p_j = 1|TEC$. The authors first proved that there exists at least an optimal solution to $1|\text{prmp}, s_j \in \mathbb{R}_+, d_j, p_j = 1|TEC$ in which the ordering of jobs satisfies the *Earliest Due Date* (EDD) rule, i.e., the property for which a job precedes another if and only if the due date of the former is less than or equal to the due date of the latter. Then, Penn and Raviv showed that such a problem can be solved in $O(NK^2)$ through dynamic programming after sorting the jobs according to their due dates. This result led the authors to conjecture the existence of an $O(NK^2)$ exact algorithm also for the $1|\text{prmp}, s_j \in \mathbb{R}_+, p_j = 1, r_j|TEC$ problem, i.e., a version of the BJSTP that considers the release dates instead of the due dates. This conjecture remains an open problem.

Fang et al. (2016) focused on versions of the BJSTP characterized by having a power demand $q_j > 0$ associated with each job $j \in \mathcal{J}$. These versions derive from the modeling of energy-intensive manufacturing processes in which the energy demands are non-negligibly job-dependent. In presence of a power demand, a job $j \in \mathcal{J}$ that is processed on a machine h during the time slots in $\mathcal{T}_j \in \mathcal{T}$ is characterized by a processing cost equal to $q_j \sum_{t\in\mathcal{T}_j} c_t$ and by a power demand $q_{j,h}$. One of the earliest versions of the BJSTP with power demands is the $1|\text{prmp}, s_j \in \mathbb{R}_+, q_j|TEC$, i.e., a version of the BJSTP in which the time horizon is partitioned into a set of time periods as in Fig. 6(c). Fang et al. (2016) proved that there is at least an optimal schedule to the $1|\text{prmp}, s_j \in \mathbb{R}_+, q_j|TEC$ such that the jobs with a higher power demand are associated with time slots with a lower cost, and showed that the optimal solution to the $1|\text{prmp}, s_j \in \mathbb{R}_+, q_j|TEC$ can be computed with a greedy polynomial-time algorithm by using arguments similar to the ones used by Penn and Raviv for the $1|\text{prmp}, s_j \in \mathbb{R}_+, d_j, p_j = 1|TEC$. The authors also studied the $1|\text{speed}, s_j \in \mathbb{R}_+, q_j, w_j|TEC$, i.e., a version of the BJSTP in which a job $j \in \mathcal{J}$, when processed at speed v, has a processing time $p_j = w_j/v$ and a power demand $q_j = v^\delta$ for some scalar $\delta > 1$. This problem considers a machine endowed with speed scaling capabilities. As a result, the problem also requires deciding the operation speed v at which the machine processes the jobs. This constitutes a non-trivial task, since a higher machine speed enables lower processing times, but entails an increase in power demands. On the other hand, a lower speed helps in reducing the TEC, at the expense of lower efficiency in production. Penn and Raviv first proved the strong $\mathcal{NP}$-hardness of this problem and provided an approximate solution algorithm characterized by an error ratio of $\sum_{k=1}^{K}(c_k / \min_{t\in\mathcal{T}} c_t)^{\delta/(\delta-1)}$ for some scalar $\delta > 1$.

Chen et al. (2021) studied three different preemptive versions of the BJSTP related to the ones proposed by Fang et al. (2016) and Che et al. (2016), namely (i) the $1|\text{prmp}|\sum_{j\in\mathcal{J}} C_j + TEC$, i.e., the problem of optimizing the sum of the completion times of the jobs and the TEC; (ii) the $1|\text{prmp, seq}, w_j|\sum_{j\in\mathcal{J}} w_jC_j + TEC$, i.e., the problem of minimizing the weighted sum of the completion times plus the TEC, while considering a fixed sequence for jobs as a problem parameter; and finally (iii) the $1|\text{prmp}|\sum_{j\in\mathcal{J}} w_jC_j + TEC$, i.e., a generalization of (ii) in which also the job sequence must be determined. For each of the three problems, Chen et al. considered a time horizon partitioned as in Fig. 6(b). By exploiting the assumption of dealing with a fixed job sequence in (ii) the authors succeeded in showing the existence of an $O(NK)$ exact solution algorithm for $1|\text{prmp, seq}, w_j|\sum_{j\in\mathcal{J}} w_jC_j + TEC$ by using a dynamic programming approach similar to the ones presented by Fang et al. and Penn & Raviv for their respective versions of the BJSTP. The existence of an optimal job sequence following the SPT first rule for problem (i) allowed the authors to prove the existence of an exact algorithm for $1|\text{prmp}|\sum_{j\in\mathcal{J}} C_j + TEC$ as well and the theoretical results obtained for $1|\text{prmp, seq}, w_j|\sum_{j\in\mathcal{J}} w_jC_j + TEC$ allowed the authors to show the existence of an $O(4+\epsilon)$-approximation algorithm also for $1|\text{prmp}|\sum_{j\in\mathcal{J}} w_jC_j + TEC$.

Some recent works on the BJSTP focused on conflicting objectives that usually arise in industrial applications when considering energy efficiency as a performance criterion. Specifically, energy-conscious production plans in general are unable to attain high productivity goals. As an example, favoring lower-cost periods of the time horizon may cause production delays, which in turn may impact measures of timeliness such as total tardiness or makespan. To address these practical issues, some authors considered versions of the BJSTP with multiple objectives. As an example, Rubaiee & Yildirim (2019) focused on the bi-objective version of the BJSTP that requires preemptively scheduling a set of independent jobs on a single machine to minimize the makespan and the TEC. The authors provided a possible *Mixed-Integer Linear Programming* (MILP) formulation to exactly solve the $1|\text{prmp}|C_{\max}, TEC$ as well as an approximate solution algorithm based on the *Ant Colony Optimization* (ACO) metaheuristic (Gendreau & Potvin, 2010).

3.1.2. Versions of the BJSTP based on time periods

Fang et al. (2016) considered the $1|s_j \in \mathbb{R}_+, q_j|TEC$, i.e., the problem of finding a single-machine schedule with the minimum TEC for a given set of jobs, in presence of (i) job power consumptions and (ii) a time horizon consisting of a set of time periods as in Fig. 6(c). The authors proved that the problem is strongly $\mathcal{NP}$-hard and inapproximable within a constant $\epsilon > 1$, unless $\mathcal{P} = \mathcal{NP}$. The authors achieved these results by using a reduction from the 3-Partition Problem similar to the one used by Wan & Qi for the problems discussed in Section 3.1.

Che et al. (2016) considered the $1|s_j \in \mathbb{R}_+, q_j|TEC$, i.e., a version of the BJSTP that involves jobs with non-integral start times and power consumption. Non-integral start times constitute a realistic assumption for production systems, where machines can start processing jobs at any time of the available horizon, provided that other technological constraints, such as equipment availability and precedence relations, are satisfied. The authors developed both an exact solution approach for the $1|s_j \in \mathbb{R}_+, q_j|TEC$ based on *Mixed Integer Linear Programming* (MILP) and a greedy heuristic to approximate its optimal solution. The proposed MILP formulation exploits both a set of 0/1 variables to determine whether a given job j (or part of it) is processed in a specific period and a set of continuous variables to model non-integral start times. This formulation proved computationally unable to tackle real instances of the problem, hence the authors proposed the greedy heuristic. The idea at the core of the greedy heuristic consists of sorting first the jobs according to the non-increasing order of their power consumption and subsequently iteratively inserting the jobs, one after another, into the available time periods by selecting the insertion that greedily minimizes the objective function. Computational experiments showed a remarkable ability of the greedy heuristic to reach close-to-optimal solutions for every instance considered in the experimental analysis. Subsequently, Zhang et al. (2018a) described some possible improvements to Che et al. (2016)'s heuristic, enabled by the use of larger neighborhoods for inserting jobs. The authors also reported the effectiveness of the heuristic in a real-world case study based on a TOU pricing scheme used in China.

Chen & Zhang (2019) studied the $1|s_j \in \mathbb{R}_+|TEC$, i.e., a version of the BJSTP that involves jobs with non-integral start times. The authors proved the strong $\mathcal{NP}$-hardness of the problem by using a reduction from the 3-Partition Problem. Subsequently, they investigated the combinatorics of the $1|s_j \in \mathbb{R}_+|TEC$. In particular, they first introduced the notion of a *valley* for a given partition of $\hat{\mathcal{T}}$, which consists of a time period whose TOU cost is lower than the one of its two neighbors in $\hat{\mathcal{T}}$, i.e., the previous and the subsequent element in the partition. The authors found that the $1|s_j \in \mathbb{R}_+|TEC$ can be solved in $O(|\hat{\mathcal{T}}|)$ if $\hat{\mathcal{T}}$ contains a single valley. Moreover, they provided an FPTAS for the problem when $\hat{\mathcal{T}}$ contains at most 2 valleys. They finally gave the expression of the computational complexity for an optimal pseudo-polynomial algorithm for the problem when the number of valleys in $\hat{\mathcal{T}}$ is greater than 2. Table 3 provides a compendium of the results achieved by the authors cited so far. It is however interesting to observe that the results obtained by Chen & Zhang again highlight how, for some versions of the BJSTP, the existence of efficient solution algorithms depends on particular properties of the sequence of TOU costs.

3.1.3. Batch scheduling versions of the BJSTP

One of the earliest results in batch scheduling versions of the BJSTP was presented by Cheng et al. (2014), who developed a bi-objective MILP formulation for the $\mathcal{NP}$-hard problem $1|\text{batch}(b)|C_{\max}, TEC$. The authors proposed a solution algorithm based on the *ϵ-constraint solution approach* for bi-objective optimization (see Haimes et al., 1971), which enables the computation of a set of non-dominated solutions to the problem if at least one of the objectives can assume only integer values. The idea at the core of this solution approach consists of iterating, over all the possible values that can be assumed by the integer objective, the resolution of an optimization problem involving the remaining objectives. When this approach is carried out exhaustively, the set of non-dominated points computed by the ϵ-constraint algorithm corresponds to the optimal Pareto front. On this rationale, for each possible value $\hat{C}$ of the makespan, the ϵ-constraint solution approach proposed by Cheng et al. iteratively solves a MILP formulation for the $1|\text{batch}(b)|TEC$, by constraining the makespan of the resulting schedule to not exceed $\hat{C}$. In Cheng et al. (2016a), the authors speeded up the solution time of their approach, by presenting preprocessing techniques able to remove a vast part of the original decision variables and constraints in their MILP formulation. The main ideas at the core of this algorithm were further used, developed, and refined in subsequent works dealing with bi-objective scheduling such as those proposed by Anghinolfi et al. (2021); Wang et al. (2018a); Wu et al. (2021).

Inspired by Cheng et al., Zhou et al. (2020) introduced a novel batch scheduling version of the BJSTP, which requires finding a schedule for a set of jobs with release dates on a single machine that simultaneously minimizes both the makespan and TEC without preemption, i.e., the $1|\text{batch}(b), r_j|C_{\max}, TEC$. The authors developed a multi-objective optimization algorithm consisting of (i) finding initial sequences of jobs through a metaheuristic (namely a PSO, see Table 2); (ii) using constructive heuristics to group the sequences of jobs into batches; and finally (iii) computing a schedule for the batches by exploiting the *Earliest Ready Time* first order. The solution so constructed eventually undergoes a local search to further improve the TEC, if possible. Zhou et al. performed an extensive experimental validation of their algorithm, which is still the state-of-the-art solution approach for the problem.

Finally, Cheng et al. (2016b) investigated the $1|\text{batch}(b)|TEC$ and proposed two different MILP formulations to solve it exactly. In particular, the authors first divided the time horizon into a set of time slots, each one having a specific duration and electricity cost. This natural modeling approach gave rise to a formulation characterized by a large number of both decision variables and constraints. In order to reduce the size of such a formulation, the authors proposed a second formulation based on a repartitioning of the time horizon into time periods, each one having a finite number of time slots with the same TOU cost. If several consecutive time slots have the same cost, these slots can be aggregated and encoded by the same decision variable, thus lowering the overall number of decision variables in the formulation. Cheng et al. showed that the second formulation always outperforms the first one, according to the computational experiments carried out on instances based on TOU prices recorded in Australia, Canada, China, and France. Cheng et al. (2016b)'s work stands out as one of the few efforts in single-machine versions of the BJSTP with batch scheduling, as it considers a single objective, differently from most of the other studies in TOU scheduling with parallel batch processing. In the remainder of the paper, we refer to a formulation that exploits a partitioning of the time horizon into time slots, e.g., Cheng et al.'s first formulation, as a *time-slot-indexed formulation*. On the contrary, we call it a *time-period-indexed formulation* if it considers the time horizon as a partition into time periods, e.g., Cheng et al.'s second formulation.

3.1.4. Versions of the BJSTP with "on/off" switching

A particular niche of the literature considers also the possibility of switching the machines on and off. Within such a setting, a machine can be in one of these three states: (a) "processing", (b) "idle", or (c) "shutdown". A machine in the state (b) requires energy despite not processing any job but is ready to start processing a job at any time. A machine in the state (c) is instead shut down, i.e., it is "off". Its energy consumption is zero, but it cannot process any job without first transitioning into the state (a). However, in order to do so, a machine requires a certain amount of time and energy, which depends on the considered scheduling model. Hereinafter, we give a brief overview of the versions of the BJSTP that involve machines endowed with "on/off" switching capabilities.

The main results in the literature on versions of the BJSTP with "on/off" switching are due to Shrouf et al. (2014) and Aghelinejad et al. (2016). Shrouf et al. (2014) first introduced the $1|\text{on/off}|TEC$ and provided both a MILP formulation to exactly solve it and a metaheuristic (namely a GA) to approximate its optimal solution. Their industry-oriented model was a milestone for the formalization of the "on/off" switching mechanism in TOU scheduling. Aghelinejad et al. (2016) built upon Shrouf et al.'s results, by providing an improved MILP formulation for the problem (Aghelinejad et al., 2016), improved lower bounds (Aghelinejad et al., 2019a), and primal heuristics (Aghelinejad et al., 2018b). Aghelinejad et al. (2017) also considered the preemptive version of the $1|\text{on/off}, \text{prmp}|TEC$ and developed a dynamic programming solution algorithm for it based on the idea of representing the problem by using a directed graph with K different levels, one for each time slot t, $1 \le t \le K$. Each level consists of different nodes in the graph, and each node represents a different possible usage for the available machine during the related time slot, e.g., idle or processing. Nodes are connected by edges whose weight consists of the cost of that transition, e.g., the cost due to switching on a machine. This idea enables transforming the original scheduling problem into a shortest path problem that can be solved in $O(K^3)$ with dynamic programming. In subsequent works, the authors introduced the assumption of a fixed processing sequence for the jobs on the machine, thus inducing a total ordering on the start times (Aghelinejad et al., 2019a). Specifically, denoting a fixed processing sequence of jobs as seq, Aghelinejad et al. (2019a) developed an $O(K^3)$ algorithm for the $1|\text{on/off}, \text{seq}, q_j|TEC$ by adapting the idea at the core of the dynamic programming algorithm used for the $1|\text{on/off}, \text{prmp}|TEC$ discussed in Aghelinejad et al. (2017). In Aghelinejad et al. (2018a), the authors further extended their work, by considering the power demands of the jobs in a speed-scalable version of the problem. Specifically, Aghelinejad et al. considered the $1|\text{on/off}, \text{speed}, \text{seq}, q_j|TEC$, and they again adapted the optimal polynomial-time dynamic programming algorithm proposed in Aghelinejad et al. (2017) to obtain a pseudo-polynomial time optimal algorithm that is able to solve the problem also in the presence of a machine with variable speed. Such an algorithm is characterized by an $O(K^2V(K+V))$ worst-case complexity, where V is the number of distinct processing speeds of the machine. It is noteworthy to observe that the assumption of a fixed ordering sequence enables some versions of the BJSTP, such as the $1|\text{on/off}, \text{seq}, q_j|TEC$ and the $1|\text{on/off}, \text{speed}, \text{seq}, q_j|TEC$, to be solved to optimality in polynomial (Aghelinejad et al., 2019a; 2019b) or pseudo-polynomial (Aghelinejad et al., 2018a) time.

3.2. Multiple-machine versions of the BJSTP

Some versions of the BJSTP consider not just a single machine, but rather a set $\mathcal{H} = \{1, 2, \dots, M\}$ of M machines (Anghinolfi et al., 2021; Ding et al., 2016). In these versions, each job $j \in \mathcal{J}$ has to be scheduled in a subset $\mathcal{T}_j \subseteq \mathcal{T}$ on a machine $h_j \in \mathcal{H}$ in order to minimize the total energy consumption. For each pair of distinct jobs $j, j' \in \mathcal{J}$, it is assumed that $\mathcal{T}_j \cap \mathcal{T}_{j'} = \emptyset$ only if j and j' are scheduled on the same machine $h_j = h_{j'}$.

A solution for TOU scheduling problems with multiple machines can be generalized from (1) as follows. We define a *job-machine-time slot assignment* as a triplet $(j, h_j, \mathcal{T}_j)$ such that j is a job in $\mathcal{J}$, h_j is a machine in $\mathcal{H}$, and $\mathcal{T}_j$ is a non-empty subset of p_j time slots in $\mathcal{T}$ during which job j is processed on the machine h_j. Then, we define a schedule S on multiple machines as

$$S = \{(j, h_j, \mathcal{T}_j) : \mathcal{T}_j \subseteq \mathcal{T}, h_j \in \mathcal{H}, \forall\, j \in \mathcal{J}, \\ \text{and } \mathcal{T}_{j'} \cap \mathcal{T}_{j''} = \emptyset \ \forall\ j', j'' \in \mathcal{J}, j' \neq j'', h_{j'} = h_{j''}\}. \quad (8)$$

Each machine $h \in \mathcal{H}$ is usually associated with a positive energy consumption rate u_h. The energy consumption rate affects the energy cost of the scheduled jobs. In particular, if job $j \in \mathcal{J}$ is scheduled in the time slots in $\mathcal{T}_j$ on machine $h_j \in \mathcal{H}$, the cost associated with the processing of the job j is $u_{h_j} \sum_{k \in \mathcal{T}_j} c_k$. Consequently, we can generalize the TEC of a parallel machines schedule as

$$E(S) = \sum_{j \in \mathcal{J}} u_{h_j} \sum_{t \in \mathcal{T}_j} c_t. \tag{9}$$

Observe that in the case $\mathcal{H} = \{1\}$ and $u_1 = 1$, then (8) and (9) reduce to (1) and (2), respectively.

In the remainder of the section, we provide a taxonomy of the literature on the versions of the BJSTP that involve multiple machines. In Section 3.2.1, we discuss the versions of the BJSTP characterized by *unrelated parallel machines*. In these versions, the considered machines may process jobs at a different speed. Each job $j \in \mathcal{J}$ is characterized by processing time $p_{j,h}$, which depends on the considered machine $h \in \mathcal{H}$ (see Pinedo, 2016 for a comprehensive review of multiple-machine environments in classical scheduling). In Section 3.2.2, we focus on versions of the BJSTP characterized by having *identical parallel machines*. These problems are a special case of unrelated parallel machines versions of the BJSTP, in which each job has the same processing time on each machine, i.e., for each $j \in \mathcal{J}$, $p_{j,h} = p_{j,h'}$ holds for each $h, h' \in \mathcal{H}$. In Section 3.2.3, we consider *flow shop* versions of the BJSTP, i.e., versions characterized by the presence of two or more machines, in which each job must be processed by every machine in the order determined by the sequence. Finally, in Section 3.2.4 we present *job shop* versions of the BJSTP. In such problems, each job has to follow a specific route through the machines in the job shop.

3.2.1. Unrelated parallel machines

One of the earliest works in TOU scheduling with multiple machines was proposed by Moon et al. (2013), inspired by the practical problems of South Korean electricity industries. The problem requires finding a schedule for a set of independent jobs on unrelated parallel machines that minimizes the weighted sum of the makespan with the TEC, i.e., $Rm||GC_{\max} + TEC$, $G > 0$. The time horizon is partitioned into a set of time slots as in Fig. 6, and the constant G, given as a part of the problem data, represents a penalty cost, such as employee overtime (Moon et al., 2013). This objective captures the performance requirements of an effective production schedule, without neglecting the energy efficiency of the process. Moon et al. proposed a MILP formulation to solve the problem exactly. Unfortunately, this formulation proved to be of little avail in practice due to its inability to solve even small instances of the problem. Hence, the authors suggested the use of metaheuristics (namely a GA) to approximate the optimal solutions to real instances of the problem.

Moon et al.'s work inspired further research on the topic in the subsequent years. For example, Cheng et al. (2019) extended Moon et al.'s MILP formulation by introducing a set of strengthening valid inequalities able to shorten its solution times up to 51.23% in the same computing environment. Kurniawan et al. (2020a) pursued the search for an effective metaheuristic algorithm for the problem, by proposing a GA that builds upon Moon et al.'s design. In (Kurniawan et al., 2017), the authors focused instead on the $Rm|s_j \in \mathbb{R}_+|GC_{\max} + TEC$, $G > 0$, i.e., a slightly different version of Moon et al.'s problem characterized by non-integral start times. Kurniawan et al. presented a MILP formulation to solve this problem exactly, by adding a new set of decision variables to Moon et al.'s formulation to handle the non-integral (i.e., continuous) start times. The authors also developed a metaheuristic for the problem (namely a GA) that proved able to solve the test instances by never exceeding an 8% optimality gap.

Ding et al. (2016) considered a version of the BJSTP that only requires the minimization of the TEC for a set of jobs characterized by machine-dependent power consumptions, i.e., the $Rm|s_j \in \mathbb{R}_+, q_{j,h}|TEC$, with a time horizon partitioned into time periods as in Fig. 6(c). The authors showed the $\mathcal{NP}$-hardness of the problem and proposed a time-slot-indexed MILP formulation that builds upon Moon et al. (2013)'s work. In more detail, Ding et al. (2016)'s formulation employs specific decision variables and constraints to handle jobs that are processed across different time periods. Such variables also allow the formulation to keep track of the amount of processing performed by jobs during each time period. Furthermore, the authors developed an exact algorithm that exploits a column generation heuristic built on a Dantzig-Wolfe decomposition of the problem based on the following observation: when each job is assigned to some of the M machines, then the original problem reduces to the M independent problems of sequencing the jobs assigned to the machines. Ding et al. exploited this intuition to decompose the problem into a master problem and M (independent) subproblems, each one corresponding to a different machine. The authors conducted an experimental campaign that highlighted that the column generation heuristic enabled an improvement of up to two orders of magnitude in the computational times with respect to the ones achieved by only using the MILP formulation.

Cheng et al. (2018) considered the same problem and improved the MILP formulation proposed by Ding et al. by using fewer decision variables. Cheng et al. achieved this result by replacing the binary variables used by Ding et al. to express the end times of jobs in the time periods with a set of binary variables that simply specify whether a job is processed by some machine. Despite the additional constraints required by the introduction of these variables, this reformulation allowed the authors to achieve higher computational efficiency than Ding et al.'s formulation.

Che et al. (2017) focused on a version of the BJSTP that is very similar to the one proposed by Ding et al. and Cheng et al., i.e., the $Rm|s_j \in \mathbb{R}_+|TEC$. In fact, such a version of the BJSTP requires the minimization of the TEC on unrelated parallel machines as in Ding et al. (2016) and (Cheng et al., 2018), but it also allows the start times of jobs to be non-integral, by considering a time horizon as in Fig. 6(c). The authors proposed a first MILP formulation and then showed some characterizing properties. Specifically, the authors provided a further formulation that solves the preemption version of the problem and showed the existence of an algorithm that converts an optimal preemptive solution obtained with such a formulation into an optimal non-preemptive one. This result allowed Che et al. to prove the redundancy of two families of constraints in the non-preemptive formulation. Finally, the authors disregarded such constraints and claimed that the resulting formulation is more compact than the one proposed by Ding et al., due to the lower number of decision variables and constraints. Che et al. also developed a two-step heuristic that first preemptively assigns the jobs to the machines to minimize the TEC, and secondly modifies all the job assignments that violate the non-preemption constraint. The authors reported very low optimality gaps (less than 2%) for the two-step heuristic with respect to the MILP formulation. Saberi-Aliabad et al. (2020) improved Che et al.'s formulation by introducing an upper bound on the maximum number of time periods that can be used to process each job. The authors also introduced a strengthening inequality that sets a subset of binary decision variables used to represent the start times of jobs to zero. Saberi-Aliabad et al.'s formulation achieved an average improvement in the computational times that amounts to 43% and 34% on two state-of-the-art datasets of instances based on Chinese and European TOU pricing schemes, respectively. Saberi-Aliabad et al. were also able to solve a slightly larger number of instances compared to Che et al..

Pei et al. (2021) were the first to consider the bi-objective version of the BJSTP that requires the optimization of the makespan and the TEC on unrelated parallel machines, i.e., the $Rm||C_{\max}, TEC$. First, Pei et al. formulated the problem as a mixed-integer quadratic program that requires the optimization of a linear combination of $C_{\max}$ and TEC. Then, the authors provided a further formulation by exploiting several lemmas, and then described a set of ad-hoc cutting planes. The authors also described an approximate algorithm based on such mathematical programming ideas, that achieved an average 4% gap with respect to the optimal solutions on the considered test instances. Recently, Rego et al. (2022) tackled the bi-objective problem $Rm|s_j \in \mathbb{R}_+, s_j^k|C_{\max}, TEC$, by fitting in a niche of the literature on unrelated parallel machine scheduling that still did not consider sequence-dependent setup times. Rego et al. proposed a bi-objective mathematical formulation and subsequently transformed it into a single-objective one by expressing the objective function as an affine combination of $C_{\max}$ and TEC. The authors embedded the latter formulation as the main component of an exact weighted sum approach that samples the optimal Pareto front by iterating over different values of the objective function parameters.

We conclude the section by mentioning Tan et al. (2019)'s work which, to the best of our knowledge, is the only one dealing with a version of the BJSTP that involves *uniform parallel machines*. Such a processing environment consists of a special case of unrelated parallel machines when each job $j \in \mathcal{J}$ is associated with a processing time p_j, each machine $h \in \mathcal{H}$ is endowed with a processing speed ν_h, and the time required for machine h to process job j is p_j/ν_h. The authors considered the problem of minimizing the TEC on a set of uniform batch parallel machines, i.e., the $Qm|\text{batch}(b)|TEC$. We can also view this version of the BJSTP as a particular case of the problem proposed by Moon et al., i.e., the $Rm||GC_{\max} + TEC$, $G > 0$, when the machines are uniform and parallel batch processing and $G = 0$. Tan et al. proposed a MILP formulation for the problem and developed a *Single-Population Genetic Algorithm* (SPGA) and a *Multi-Population Genetic Algorithm* (MPGA). Such two metaheuristics are GAs that employ a different way of enhancing and preserving diversity in the population of solutions. The authors compared the SPGA and the MPGA implementation with the MILP formulation to evaluate their optimality gap, which never exceeded 10% on the considered test instances.

3.2.2. Identical parallel machines

Wang et al. (2018a) first introduced the $Pm||C_{\max}, TEC$, i.e., a multi-objective version of the BJSTP on parallel identical machines, in the literature on TOU scheduling. The authors developed a heuristic to approximate the optimal solution to the problem, and also presented a possible MILP formulation. Wang et al.'s heuristic is based on the ϵ-constraint paradigm for bi-objective optimization (see Section 3.1 and Haimes et al., 1971). The heuristic computes a set of non-dominated solutions by iterating over the $O(K - C_{\min} + 1)$ possible values for the makespan, where the expression for $C_{\min}$ is similar to the lower bound provided by Pinedo (2016) for the optimization of the makespan on identical parallel machines. At each iteration, for a given makespan value $\hat{C}$, Wang et al.'s heuristic computes a schedule that minimizes the TEC and whose makespan does not exceed $\hat{C}$. First, a constructive heuristic considers the jobs according to the *Longest Processing Time* (LPT) first order, which sorts jobs in non-increasing order of their processing times. Then, it assigns each job to the set of adjacent slots with the smallest cost that (i) have not been assigned to any other job yet, and (ii) satisfy the constraint on $\hat{C}$, i.e., none of the slots is greater than $\hat{C}$. If the constructive heuristic computed a feasible solution, then a local search tries to improve its TEC by shifting blocks of assigned slots on each machine separately. Observe that, due to its greedy approach, the constructive heuristic may fail to find a feasible solution even though there exists one. Finding such a feasible solution corresponds to the problem of determining whether there exists a schedule for a set of jobs satisfying a maximum makespan constraint, which is an $\mathcal{NP}$-complete problem (Pinedo, 2016). The MILP formulation proposed by Wang et al. adapts Che et al.'s time-period-indexed formulation to the case of identical parallel machines. However, as subsequently highlighted by Anghinolfi et al. (2021), Wang et al.'s formulation sometimes allows machines to process two jobs at the same time, despite being single-job processing. As a result, Wang et al.'s formulation is not valid for each possible instance of the problem.

Anghinolfi et al. (2021) advanced on the combinatorics of the $Pm||C_{\max}, TEC$ by building upon Wang et al.'s results described in the previous paragraph. Specifically, Anghinolfi et al. first proposed a novel MILP formulation for the problem that overcomes the drawbacks of Wang et al.'s formulation. Furthermore, the authors proposed a heuristic based on the ϵ-constraint paradigm (first discussed in Section 3.1.3), consisting of a constructive heuristic called Split-greedy Heuristic and a novel local search named Exchange Search. Split-greedy Heuristic considers the jobs according to the LPT first order, similarly to Wang et al.'s constructive heuristic. However, at each iteration, it greedily assigns a job $j \in \mathcal{J}$ to a subset of idle slots $\mathcal{Z} \subseteq \mathcal{T}$ that are either (i) consecutive, i.e., $|\mathcal{Z}| = p_j$, or (ii) non-consecutive, i.e., $|\mathcal{Z}| > p_j$, provided that the resulting preemptive schedule can be converted into a non-preemptive one with the same makespan and TEC. This is the case of a set $\mathcal{Z} = \{t_1, \ldots, t_n\} \subseteq \mathcal{T}$, $n > p_j$, of non-consecutive idle slots on some machine such that all the slots in between, i.e., $\forall t' : t' \notin \mathcal{Z}, t_1 < t' < t_n$, are already assigned to other jobs. In this way, at the end of the iterations, a (possibly) preemptive schedule is converted into a non-preemptive one with the same makespan and TEC. Split-greedy Heuristic also enables to increase the quality of the computed solutions with respect to Wang et al.'s by using larger neighborhoods for the greedy choices. The purpose of Exchange Search is to improve the TEC of the schedule computed by Split-Greedy Heuristic. In particular, given a schedule S, Exchange Search iteratively considers pairs of single-machine subschedules of S, and reschedules jobs from one subschedule to the other in order to locally improve the TEC. If such a move is improving, then Exchange Search updates S accordingly. Exchange Search stops iterating when no further improvement in TEC is possible. A complete explanation of Exchange Search requires several additional notions. We refer the interested reader to Anghinolfi et al. (2021) and Ronco (2022). Anghinolfi et al. experimentally showed that their heuristic scheme outperformed Wang et al.'s heuristic on the original Wang et al.'s test instances, as well as a further set of larger instances created for additional testing.

Ronco (2022) provided a further improvement on the representation of the solution space for the problem. Specifically, with a little abuse of notation, we denote by $\mathcal{P} = \{p_j : j \in \mathcal{J}\}$ the set of distinct processing times in $\mathcal{J}$, and by $\mathcal{J}_d = \{j : p_j = d\}, d \in \mathcal{P}$, the subset of jobs in $\mathcal{J}$ with processing time equal to d. Ronco proved that for each feasible solution S, there exist at least other $\prod_{d \in \mathcal{P}} |\mathcal{J}_d|! - 1$ feasible solutions with the same makespan and TEC of S. The proof is based on the observation that exchanging the assignments of two jobs with the same processing time in a schedule for the problem does not alter its makespan, nor its TEC. In particular, while Anghinolfi et al.'s formulation used the binary decision variables $X_{j,h,t}, j \in \mathcal{J}, h \in \mathcal{H}, t \in \mathcal{T}$, where $X_{j,h,t}$ equals 1 if job j starts at time slot t on machine h, and 0 otherwise, Ronco's formulation employed the binary decision variables $Y_{d,h,t}, d \in \mathcal{P}, h \in \mathcal{H}, t \in \mathcal{T}$, where $Y_{d,h,t}$ equals 1 if a job with processing time d starts at time slot t on machine h, and 0 otherwise. Observe that while the former variables explicitly consider each job, the latter implicitly distinguish jobs exclusively according to their processing times. Ronco showed that there exists a valid formulation for the problem expressed in terms of the $Y_{d,h,t}$ variables. From an experimental viewpoint, such a formulation reflected into a significantly lower computational burden both in terms of memory requirements and computational times on the considered test instances (the same used by Anghinolfi et al.).

Zhou et al. (2018) expanded the problem considered by Wang et al. and Anghinolfi et al. to a batch scheduling version of the BJSTP with release times. Specifically, Zhou et al. considered the bi-objective problem of finding a schedule for a set of jobs with release times on identical parallel batch machines that simultaneously minimizes the makespan and the TEC, i.e., $Pm|\text{batch}(b), r_j|C_{\max}, TEC$. The authors proposed a multi-objective discrete differential EA, adapted from the traditional design for continuous domains. Qian et al. (2020) considered the simpler problem $Pm|\text{batch}(b)|C_{\max}, TEC$, by disregarding the release times. The authors developed a multi-objective EA that fully exploits a clustering method able to extract information on the distribution of solutions in the search space. Differently, Rocholl et al. (2020) considered the minimization of the total weighted tardiness in place of the makespan, i.e., the $Pm|\text{batch}(b), r_j|\sum_{j\in\mathcal{J}} w_jT_j, TEC$, and proposed a MILP formulation as well as heuristic approaches for the general case of the problem. The authors also presented several observations leading to a more compact formulation of a special version of the problem where all the release times are equal to zero.

3.2.3. Flow shops

Luo et al. (2013) focused on one of the first flow shop versions of the BJSTP. Specifically, the authors considered the problem of minimizing the makespan and the TEC in a flow shop, i.e., the $FFc||C_{\max}, TEC$. The authors proposed a multi-objective metaheuristic based on ACO (first discussed in Section 3.1.1) which paved for all the subsequent applications of metaheuristics to flow shop versions of the BJSTP. Inspired by Luo et al., Zhang et al. (2019) focused on another flow shop version of the BJSTP that requires the simultaneous optimization of the makespan and the TEC for a set of independent jobs on machines endowed with the “on/off” switching (first discussed in Section 3.1.4). Zhang et al. proposed a MILP formulation that highlighted how a minimum makespan schedule for the problem can be achieved at the expense of a 45% increase in the minimum TEC. Wang et al. (2020b) further investigated Zhang et al.’s version of the BJSTP by proposing a two-machine *flexible flow shop*, where the two machine stages consist of a set of parallel identical machines followed by a batch machine. A flexible flow shop generalizes a flow shop by considering one or more processing stages, and each job has to be processed by one and only one machine in a stage before proceeding to the next one (Pinedo, 2016). Similarly to Zhang et al., Wang et al. considered the “on/off” switching mechanism as well as the makespan and the TEC as optimization objectives. Zhang et al. also considered the $FF2(Qm, Rm')||TEC$, $m > m'$, i.e., a two-machine flexible flow shop where the first and the second stage consist of m uniform machines and m' unrelated machines, respectively, and the number of machines in the first stage is always greater than the one in the second stage, i.e., $m > m'$ (Zhang et al., 2021). Finally, Chen et al. (2022) proposed a flexible flow shop version of the BJSTP that considers ladder energy pricing along with TOU energy prices. Ladder energy pricing, typically applied for energy-intensive industries to control carbon emissions, adds to the existing pricing by further burdening the consumer with a non-linear function of the energy costs (Dobbs, 2015). Chen et al. proposed a specific implementation of NSGA II, endowed with a heuristic for the generation of the initial population and a problem-specific adaptive genetic operator to ensure diversity and quality in the individuals.

Cheng et al. (2016c) proposed a simple MILP formulation for the $Fm|\text{prmu}|TEC$, i.e., the problem of minimizing the TEC in a *permutation flow shop*, i.e., a flow shop version of the BJSTP in which each machine processes jobs according to the *First-In First-Out* policy. This policy enforces the processing order to be the same as the arrival order of the jobs in the processing queue. Similarly to Cheng et al., Wang et al. (2018b) proposed a MILP formulation for a two-machine permutation flow shop version of the BJSTP with the usual objective of minimizing the TEC, i.e., the $F2|\text{prmu}|TEC$. The authors developed a heuristic based on Johnson’s rule and dynamic programming, and provided a polynomial-time algorithm when the job sequence is fixed. Ho et al. (2021) extended Wang et al.’s work by proposing a novel formulation and a heuristic for the same problem. Recently, Ho et al. (2022) achieved further results on the problem, by adding the “on/off” switching mechanism to the machines and proposing an exact algorithm based on MILP and Benders decomposition. Instead, Wang et al. (2020a) described a MILP formulation for a generalization of Cheng et al.’s problem that also considers the makespan as an optimization objective. Similarly, Kurniawan & Fujimura (2020) considered a general permutation flow shop version of the BJSTP that requires the minimization of the TEC and the total weighted tardiness, instead of the makespan, i.e., the $Fm|\text{prmu}, w_j|\sum_{j\in\mathcal{J}} w_jT_j, TEC$. The authors developed an implementation of the *Strength Pareto-archived Evolutionary Algorithm* (SPEA) metaheuristic (Kim et al., 2004), which improves upon the classical evolutionary design by periodically archiving solutions. This strategy allows the preservation of individuals with high fitness values from further crossover or mutation, while simultaneously increasing quality and diversity of the population of solutions.

Badri et al. (2021) disregarded Cheng et al.’s permutation component of the flow shop, but they added the minimization of the sum of the total earliness and tardiness as a further objective, i.e., the authors considered a version of the BJSTP that can be represented as the $Fm||TEC, \sum_{j\in\mathcal{J}} E_j + T_j$. Badri et al. (2021) first proposed a bi-objective MILP formulation for the problem. Then, the authors claimed the existence of an equivalent single-objective MILP formulation, and proposed a fuzzy multi-objective approach to convert the proposed bi-objective formulation into a single-objective one (Kannan et al., 2013). Finally, Aghelinejad et al. modeled a two-machine flow shop scheduling version of the BJSTP with the “on/off” switching mechanism, i.e., the $F2|\text{on/off}|TEC$, through MILP (Aghelinejad et al., 2020a; 2020b), and Peng et al. (2021) tackled $Fm||TEC$ with a MILP formulation and a *Particle Swarm Optimization* (PSO) metaheuristic (Gendreau & Potvin, 2010; Kennedy & Eberhart, 1995). This algorithm, originally conceived for non-linear optimization, can be used for combinatorial optimization by employing appropriate problem encodings (Gendreau & Potvin, 2010).

3.2.4. Job shops

In this subsection, we deal with the few job shop scheduling studies in the literature on the versions of the BJSTP. Among these, Kurniawan et al. (2020b) considered a bi-objective job shop version of the BJSTP requiring to minimize the TEC and the total weighted tardiness, i.e., $Jm||\sum_{j\in\mathcal{J}} w_jT_j, TEC$. The authors decomposed the problem into the subproblems of sequencing the different operations on the machines and determining their start times. They proposed a local search based on a genetic algorithm that encodes the operations sequence and their start time into two different gene representations. Li et al. (2018) presented a MILP formulation for a batch scheduling version of the BJSTP with the minimization of the makespan and TEC, i.e., $Jm|\text{batch}(b)|C_{\max}, TEC$.

Park & Ham (2022) proposed a flexible job shop with the single objective $GC_{\max} + TEC$, $G > 0$, originally introduced by Moon et al. (see Section 3.2.1), and developed an IP formulation and a constraint programming approach to achieve the two following results. First, they compared the two solution approaches and established that constraint programming attains better performances than IP. Secondly, they performed a sensitivity analysis that compared the solution computed by the constraint programming approach applied to the problem, i.e., $FJc||GC_{\max} + TEC$, $G > 0$, with the ones obtained by only minimizing the makespan, i.e., by solving $FJc||C_{\max}$. Park & Ham showed that the optimization of the former objective, i.e., $GC_{\max} + TEC$, $G > 0$, enables significant energy cost savings, with respect to the energy-oblivious $C_{\max}$ objective. Jiang & Wang (2020) considered the flexible job shop version of the BJSTP of simultaneously minimizing the makespan and TEC, and presented both a MILP formulation and a hybrid evolutionary algorithm. Such a metaheuristic employs specific operators to generate new solutions by exploiting information in their neighborhoods, and it also embeds two different intensification operators to further improve the quality of the best solutions in the population.

Table 4 reports a compendium of the problems without an exact solution algorithm that runs in polynomial or pseudo-polynomial time, according to the machine environment and the number of objective functions.

4. Non-standard versions of the BJSTP

In this section, we review versions of the BJSTP that do not fit the above classification, due to the presence of specific objective functions, structural features of the manufacturing system, or industrial scopes. In Section 4.1, we consider versions of the BJSTP that do not obey the three-field notation, because of the presence of costs related to overnight shifts or emission of particular pollutants other than CO_2. In Section 4.2, we report on versions of the BJSTP that occur in manufacturing environments that have specific functions or maintenance requirements, such as planned shutdowns or disruptive faults. Finally, in Section 4.3, we present some practical case studies of TOU scheduling.

4.1. Versions of the BJSTP with non-standard objectives

The literature on the BJSTP reports on versions of the problem whose objectives may include specific measures related to revenues and profits, or emissions. In the next subsections, we will review both of these specific two classes.

4.1.1. Objectives including measures related to revenues and profits

In the context of revenues and profits, Penn & Raviv (2021) first considered a non-preemptive single-machine version of the BJSTP that requires selecting a set of the available jobs and finding a schedule that maximizes the profit, expressed as the difference between the total revenue of the scheduled jobs and the TEC. Penn & Raviv proved that the problem is strongly $\mathcal{NP}$-hard by using a reduction from the decision version of the *bin packing problem* (Garey & Johnson, 1990). Subsequently, the authors also proved that their problem can be solved in polynomial time if the non-preemption constraint is relaxed, as it occurs in some of the problems presented in Section 3. Specifically, the authors observed that, in an optimal solution to the problem, a time period is either completely used for processing or not used at all, with the possible exception of one and only one period characterized by partial utilization. This property enables the construction of an optimal solution to the problem obtained by greedily sorting time periods in non-decreasing order with respect to their cost.

Lee et al. (2017) observed that the timeliness of production constitutes an indirect indicator of current and future income. This fact led the authors to consider a version of the BJSTP characterized by having a single machine endowed with an "on/off" switching, jobs to be processed dynamically, and an objective consisting of minimizing the sum of the TEC and the *Just-In-Time* (JIT) cost of the schedule. The JIT cost is expressed as a linear combination of the mean squared earliness and the tardiness of the jobs. The authors proposed a heuristic solution algorithm for this problem based on dynamic control and showed, through extensive computational experiments carried out on real-world instances, that considering the timeliness of production can decrease the energy consumption costs from 63% to almost 100%.

4.1.2. Objectives including measures related to emissions

In the last decades, most European nations and many North American states introduced sanctions that target the carbon content in greenhouse gases (Trevino-Martinez et al., 2022), in an effort of containing the amount of CO_2 in the atmosphere. In this sense, preventing the use of non-renewables, or restraining it to the best extent possible, has become a crucial aspect in manufacturing. Chen et al. (2019) addressed such concerns by developing a MILP formulation for an *order acceptance* version of the BJSTP that considers carbon emission costs besides TOU prices. In such a version, each job is associated with a revenue. Order acceptance scheduling requires to *accept* a subset of the available jobs as *orders* to be scheduled, so as to maximize the total revenue. This objective can be possibly combined with other productivity measures, e.g., the makespan or the total tardiness. In particular, the objective of the version of the BJSTP proposed by Chen et al. also considers both the TOU prices and the carbon emission costs. Chen et al. were able to solve only small instances, i.e., characterized by no more than 20 jobs, to optimality by using their model.

Zhang et al. (2014) considered a particular bi-objective flow shop version of the BJSTP that involves the minimization of both the TEC and the carbon emissions. Zhang et al. developed a MILP formulation that extends Moon et al.'s formulation (discussed in Section 3.2.1) by introducing carbon emissions as a further objective function and handling the different product types in the flow shop. The authors empirically showed, with some real-world tests, that the solution provided by a commercial MILP solver enables improvements in energy costs at the expense of minimal carbon emissions. Their approach is able to compute the whole Pareto optimal front for a day of operational service, to offer a comprehensive view of the trade-off between productivity and environmental measures. For instance, the authors showed that an improvement of 6.9% in TEC corresponds to very low carbon emissions in practice.

Gong et al. (2016) considered the problem of scheduling a set of manufacturing tasks with due dates on a single machine with an "on/off" switching mechanism, to minimize the linear combination of the TEC with the affine combination of the costs due to the "on/off" switching of the machine in the different work shifts. Gong et al. first gave a MILP formulation for the version of the BJSTP at hand. Then the authors developed a GA implementation as a solution approach for the problem, which they tested on a real-life case study involving a surface grinding machine and real-life measurements of energy prices. The GA was able to achieve a 52% reduction on average in the maximum energy cost per hour, along with a 19% saving in the average costs over the whole time horizon.

More recently, Gong et al. (2019) proposed a flexible job shop version of the BJSTP with job recirculation and operation sequence-dependent setups, while considering the cost of work shifts and machine working times as well. The problem requires the optimization of five different objectives: the makespan, the TEC, the total labor cost, the maximal workload, and the total machines' workload. In particular, the total labor cost considers different wages according to the time of the day for different personnel types, while the maximal workload and the total workload consist of the maximum working time among the machines and the sum of such working times, respectively. The introduction of the machine workload as a component of the objective aims to limit

Table 4
Integer programming algorithms, approximation algorithms, heuristics and metaheuristics for different versions of the BJSTP.

Class	Problem	Article	Solution approach
Single-objective single-machine scheduling	$1\|\text{prmp}, w_j\| \sum_{j\in\mathcal{J}} w_jC_j + TEC$	Chen et al. (2021)	PTAS
	$1\|s_j \in \mathbb{R}_+\|TEC$	Chen & Zhang (2019)	FPTAS ($\hat{\mathcal{T}}$ has at most 2 valleys)
	$1\|s_j \in \mathbb{R}_+, q_j\|TEC$	Che et al. (2016)	MILP, Greedy insertion heuristic
	$1\|s_j \in \mathbb{R}_+, q_j\|TEC$	Zhang et al. (2018a)	MILP, Greedy insertion heuristic
	$1\|\text{speed}, s_j \in \mathbb{R}_+, q_j, w_j\|TEC$	Fang et al. (2016)	$\sum_{k=1}^{K}(c_k\,/\,\min_{t\in\mathcal{T}} c_t)^{\delta/(\delta-1)}$-approximation algorithm, $\delta > 0$, when $p_j = w_j/v$ and $q_j = v^{\delta}$, being v the processing speed
	$1\|\text{on/off}\|TEC$	Shrouf et al. (2014)	MILP, GA
	$1\|\text{on/off}\|TEC$	Aghelinejad et al. (2016)	MILP
	$1\|\text{on/off}\|TEC$	Aghelinejad et al. (2018b)	MILP, Heuristic, GA
	$1\|\text{on/off}\|TEC$	Aghelinejad et al. (2019a)	New lower bounds for MILP formulation
	$1\|\text{batch}(b)\|TEC$	Cheng et al. (2016b)	MILP
Multi-objective single-machine scheduling	$1\|\text{prmp}\|C_{\max}, TEC$	Rubaiee & Yildirim (2019)	MILP, ACO-based algorithms
	$1\|s_j \in \mathbb{R}_+\| \sum_{j\in\mathcal{J}} w_jT_j, TEC$	Kurniawan et al. (2018)	MILP, Hybrid GA with random insertion
	$1\|\text{batch}(b)\|C_{\max}, TEC$	Cheng et al. (2014)	MILP
		Cheng et al. (2016a)	MILP, exact ϵ-constraint algorithm
		Wu et al. (2021)	MILP, Heuristics
	$1\|\text{batch}(b), \text{speed}\|C_{\max}, TEC$	Zhang et al. (2018b)	MILP, Heuristics
	$1\|\text{batch}(b), r_j\|C_{\max}, TEC$	Zhou et al. (2020)	MILP, Hybrid multi-objective meta-heuristic algorithm
Multi-objective parallel identical machines	$Pm\|\|C_{\max}, TEC$	Wang et al. (2018a)	MILP, Heuristic, GA
		Anghinolfi et al. (2021)	MILP, "Split-greedy" heuristic, "Exchange Search" local search
		Ronco (2022)	MILP, "Split-greedy" heuristic, "Exchange Search" local search
	$Pm\|\text{batch}(b), r_j\|C_{\max}, TEC$	Zhou et al. (2018)	MILP, Multi-objective differential EA
	$Pm\|\text{batch}(b)\|C_{\max}, TEC$	Qian et al. (2020)	MILP, Multi-objective EA based on adaptive clustering
	$Pm\|\text{batch}(b), r_j\| \sum_{j\in\mathcal{J}} w_jT_j, TEC$	Rocholl et al. (2020)	MILP, NSGA II with embedded heuristics
Uniform parallel machines	$Qm\|\text{batch}(b)\|TEC$	Tan et al. (2019)	MILP, SPGA, MPGA
Single-objective unrelated parallel machines	$Rm\|s_j \in \mathbb{R}_+\|TEC$	Che et al. (2017)	MILP, Two-stage heuristic
		Saberi-Aliabad et al. (2020)	MILP, fix-and-relax heuristic
	$Rm\|s_j \in \mathbb{R}_+, q_{j,h}\|TEC$	Ding et al. (2016)	MILP, Exact algorithm based on MILP
		Cheng et al. (2018)	MILP
	$Rm\|\|GC_{\max} + TEC,\ G > 0$	Moon et al. (2013)	MILP, Hybrid GA
		Cheng et al. (2019)	MILP
		Kurniawan et al. (2020a)	"Triple-chromosome" GA
	$Rm\|s_j \in \mathbb{R}_+\|GC_{\max} + TEC,\ G > 0$	Kurniawan et al. (2017)	MILP, GA
	$Rm\|\text{prmp}, s_j \in \mathbb{R}_+\|C_{\max} + TEC$	Chen et al. (2021)	Exact algorithm based on IP
Multi-objective unrelated parallel machines	$Rm\|\|C_{\max}, TEC$	Pei et al. (2021)	MILP, Approximation algorithm
	$Rm\|s_j \in \mathbb{R}_+, s_j^k\|C_{\max}, TEC$	Rego et al. (2022)	MILP-based weighted sum method, GA
Single-objective flow shops	$F2\|\text{prmu}\|TEC$	Wang et al. (2018b)	MILP, ILS
		Ho et al. (2021)	Heuristics
	$F2\|\text{on/off}\|TEC$	Aghelinejad et al. (2020a)	Two Mixed-Integer Programming formulations
		Aghelinejad et al. (2020b)	Linear Programming formulation
	$F2\|\text{on/off, prmu}\|TEC$	Ho et al. (2022)	Exact MILP algorithm with Benders decomposition
	$Fm\|\|TEC$	Peng et al. (2021)	IP, PSO
	$Fm\|\text{prmu}\|TEC$	Cheng et al. (2016c)	MILP, GA
	$FF2(Qm, Rm')\|\|TEC, m > m'$	Zhang et al. (2021)	MILP, TS - Greedy insertion algorithm
	$FFc\|\|C_{\max}, TEC$	Luo et al. (2013)	ACO-based metaheuristic
Multi-objective flow shops	$Fm\|\|TEC, \sum_{j\in\mathcal{J}} E_j + T_j$	Badri et al. (2021)	MILP
	$Fm\|\text{prmu}, w_j\| \sum_{j\in\mathcal{J}} w_jT_j, TEC$	Kurniawan & Fujimura (2020)	SPEA-based metaheuristic
	$Fm\|\text{prmu}\|C_{\max}, TEC$	Wang et al. (2020a)	MILP
	$FF2(Pm, \text{batch}(b))\|\text{on/off}\|C_{\max}, TEC$	Wang et al. (2020b)	MILP, Constructive heuristic with local search, TS, ACO
	$FFc\|\text{on/off}\|C_{\max}, TEC$	Zhang et al. (2019)	SPEA
	$FFc(Qm)\|d_j\| \sum_{j\in\mathcal{J}} T_j, TEC$	Ding et al. (2021)	MILP, Hybrid PSO
	$FFc\|\|C_{\max}, TEC$	Chen et al. (2022)	MILP, NSGA II
Single-objective job shops	$FJc\|\|GC_{\max} + TEC,\ G > 0$	Park & Ham (2022)	IP, Constraint Programming
Multi-objective job shops	$Jm\|\| \sum_{j\in\mathcal{J}} w_jT_j, TEC$	Kurniawan et al. (2020b)	LS based on GA
	$Jm\|\text{batch}(b)\|C_{\max}, TEC$	Li et al. (2018)	MILP
	$FJc\|\|C_{\max}, TEC$	Jiang & Wang (2020)	MILP, Hybrid evolutionary algorithm

Table 5
A compendium of non-standard objectives in versions of the BJSTP.

Non-standard objective	Literature
Total profit	Penn & Raviv (2021)
Just-in-time cost	Lee et al. (2017)
Carbon emissions cost	Chen et al. (2019); Trevino-Martinez et al. (2022); Zhang et al. (2014)
Labor costs	Gong et al. (2019)
Maximal and total workload of machines	Gong et al. (2019)

the number of idle machines that burden the overall power consumption without actively contributing to production. The authors developed a metaheuristic approach (namely a GA) to solve this problem, and observed that it was able to effectively scale to large instances, up to a real-world application case based on a plastic bottle manufacturer in Belgium.

Table 5 reports a summary of the non-standard objectives considered in this subsection as a part of versions of the BJSTP.

4.2. Versions of the BJSTP that include technological and maintenance issues

Real-life versions of the BJSTP often include practical aspects that may range from the technological characteristics of the machines to breakdowns and maintenance issues.

Wang (2020) first included some technological aspects in the BJSTP, by considering a specific version of the problem in which the TEC of a set of independent jobs with release dates and deadlines must be minimized on a single machine with a special requirement for job processing. Specifically, such a machine has to undergo a *calibration phase* to be capable of processing jobs for some fixed interval. In the version of the BJSTP considered by Wang, calibration cannot be performed more than a fixed number of times. By studying the combinatorics of this version, the authors succeeded in showing that there is an optimal solution to this problem that follows the *Earliest Deadline First* order, according to which a job with a smaller deadline precedes any other job with a larger deadline. As a result, the authors were able to provide the first pseudo-polynomial time dynamic programming optimal algorithm for the version of the BJSTP. Furthermore, the authors studied the same version of the BJSTP without calibration, to find out that it admits a polynomial-time greedy optimal algorithm.

Dealing with machine maintenance constitutes a necessary condition for successful, real-life production processes in the long term. In fact, careful scheduling plans that consider machine reliability and maintenance services decrease the average downtime, thus lessening its disruptive effects (Raza & Hameed, 2021). In this context, Sin & Chung (2020) enhanced the study of Shrouf et al. (2014) presented in Subsection 3.1 by considering the minimization of the TEC and the machine availability for a schedule of a set of independent jobs on a single machine endowed with an "on/off" switching mechanism and preventive maintenance requirements. Sin & Chung assumed a specific closed-form expression for the probability distribution of having the machine functioning at any given time and presented a MINLP formulation that, according to the authors, is only suitable for small-scale instances. To solve large instances of the problem, the authors also proposed a metaheuristic based on GA, which they tested on a set of instances based on Shrouf et al.'s dataset. Sin & Chung found out that this algorithm is indeed able to solve larger instances than the exact approach enabled by the MINLP formulation. In addition to this scaling capability, the accuracy of their algorithm was also higher than the baseline GA implementation for the problem on most of the test instances.

Cui et al. (2020) proposed a single-machine version of the BJSTP with maintenance requirements and an objective consisting of minimizing both the makespan and the TEC. Cui et al. proposed an algorithm that separately optimizes the makespan by a means of a GA, and then minimizes the TEC by means of branch-and-bound while satisfying maintenance constraints. The authors performed an experimental analysis that highlighted how, on the considered test instances, the proposed algorithm enables up to 35% improvement in TEC, at the expense of a slight increase in makespan. In Cui & Lu (2021), the authors further extended their algorithmic approach to a version of the BJSTP characterized by a flow shop instead of a single machine.

Finally, Kong et al. (2021) considered a version of the BJSTP that requires the minimization of the TEC on identical parallel machines, subject to dynamic disruptions and deterioration effects. Kong et al. modeled the disruptions as new "arrival" jobs to be rescheduled, alongside the "original" ones. The authors developed an ad-hoc implementation of the *Variable Neighborhood Search* (VNS) metaheuristic (Gendreau & Potvin, 2010). The idea at the core of VNS lies in adaptively changing solution neighborhoods while transitioning between the intensification phase, i.e., the search for a local optimum, and the perturbation phase, i.e., the attempt of the algorithm to escape the local optimum to possibly find better solutions. Kong et al.'s implementation exploits three different neighborhood structures. The first neighborhood considers the swaps between any two jobs scheduled on different machines; the second one evaluates the rescheduling of a job to another part of the time horizon, possibly on another machine; finally, the third one considers the possible permutations of the assignments of a set of jobs scheduled on two different machines. The ad-hoc VNS implementation was able to outperform the baseline VNS as well as other state-of-the-art metaheuristic implementations on 309 out of the 360 considered test instances.

4.3. Applications cases

This subsection describes some successful efforts in modeling industry applications as versions of the BJSTP. In this sense, the literature offers valuable guidance to practitioners in taking advantage of TOU pricing schemes to reduce energy expenses and preserve production objectives at the same time.

Hadera et al. (2015) focused on the problem of computing a one-day power-intensive schedule for a set of tasks performed by a melt shop section of a stainless steel plant. Hadera et al. were interested in minimizing both the TEC and the total weighted sum of the start times of the tasks. In this BJSTP version, plant owners can power their production by exploiting on-site generation, or purchasing and selling electricity at any given time. The authors tackled the problem by proposing a MILP formulation and a heuristic that consists of two different steps. The first step only fixes the values for the subset of variables related to sequencing. The second step exploits the information provided by the previous level to compute the actual solution by determining the start time of the jobs. Such two steps are part of an iterative loop that progressively adds integer cuts to reduce the solution space according to the solutions computed at the previous iterations. The authors showed that the joint optimization of the production and the TEC may mitigate the effects of the fluctuation of the energy prices which often occurs throughout the day in real-life applications of TOU schemes.

Cao et al. (2021) provided another example of iron-steel plant scheduling optimization. Specifically, Cao et al. described the bi-objective version of the BJSTP of simultaneously minimizing the makespan and TEC for the flow shop production planning of an iron-steel plant. In this version of the BJSTP, the TEC also depends on both the self-generation electricity costs and the on-grid electrovalence. The authors presented a multi-objective MILP formulation inspired by the work of Ding et al. (see Section 3.2) as well as a population-based metaheuristic to approximate the optimal solution to the problem. Such an algorithm exploits a hybrid crossover operator that aims at finding a better balance between the preservation and improvement of good solutions, and the discovery of new promising ones in the search space. The use of this operator enabled Cao et al.'s metaheuristic to achieve a significant reduction in the TEC and the makespan of the solutions computed for the test instances in the experimental campaign.

Zhao et al. (2018) faced a multi-stage production problem of a rolling process of steel within an electric grid that requires minimizing the total production costs, given as the sum of the costs of materials changeover, rolling facilities, and the TEC. Rolling is a refinement process, where semi-finished products are subject to multiple transformation phases up to the final polish. Zhao et al. first provided a *Mixed-Integer Non-Linear Programming* (MINLP) formulation with generalized disjunctive programming (Balas, 2018) constraints. Then the authors proposed a MILP reformulation by linearizing the MINLP formulation and also introduced a tighter lower bound for the objective. The authors showed the effectiveness of their solution approach on a case study by highlighting an average 25% improvement in TEC with respect to an energy-oblivious approach to the same production problem.

Tan et al. (2017) modeled hot steel rolling as a job-shop batch scheduling problem. Specifically, Tan et al. considered the minimization of the TEC and the optimization of a production quality measure that penalizes jumps between adjacent slabs in the rolling units. The authors developed a GA to compute a set of non-dominated solutions and proposed a heuristic approach to select the most favorable solution that lies on the Pareto front according to some input preference parameters. The experimental results obtained by the authors showed the robustness of the algorithm in reducing energy expenses in many instances, with minimal jumps across slabs in the same unit.

Yang et al. (2018) modeled a steel production process affected by uncertainty, by considering stochastic metal elements concentration in scrap steel charge within the production scheduling of a scrap steel melt shop. Scrap steel is a partially recyclable resource, as it can be used by steel plants in re-melting processes. Yang et al. proposed a robust optimization approach with two deterministic counterparts obtained by bounding the set of uncertain parameters according to a variation range. The authors experimentally validated the two deterministic solution approaches by comparison with a two-stage optimization algorithm. Such an algorithm first optimizes the overall charge according to steel demand and then minimizes the TEC and setup costs for production. Guirong & Qiqiang (2017) also considered uncertain data in scheduling, but as a part of the steelmaking-continuous casting production. The authors focused on the three-stage process of steelmaking, refining, and continuous casting. The considered problem consists of a special flow shop with special routes for the involved materials. The authors tackled the problem with an implementation of the *cross-entropy optimization algorithm* based on Monte Carlo simulation (Deng, 2006). Such an algorithm can be applied both to non-convex optimization problems with convex bounded domains and to combinatorial optimization problems by exploiting Markov Chains (Gagniuc, 2017). A detailed explanation of the cross-entropy optimization algorithm can be found in Rubinstein (1999). The experimental results showed that Guirong & Qiqiang's solution approach is able to compensate for the drop in temperature of a load due to transportation between processing stages, and to ensure high throughput at relatively low energy costs.

Pan et al. (2019) considered a full steelmaking-refining-continuous casting scheduling problem with TOU prices and electrical load tracking. The latter is a technique used for energy-intensive industries to improve the performance of the power grid by exploiting the knowledge of previous load fluctuations. Pan et al. introduced a MINLP formulation and proposed an improved SPEA metaheuristic that exploits the arithmetic operator for solutions crossover, and hybrid self-adaptive mutation operators for solution mutation. Pan et al. (2022) enhanced both Guirong & Qiqiang (2017) and Pan et al. (2019) by addressing the integration of multi-position refining furnaces in steelmaking-continuous casting scheduling. Instead of simply considering furnaces as buffers, Pan et al. took in to account that they can be used to adjust the temperature and the composition of the molten steel. The authors developed a subgradient algorithm that exploits the Lagrangian relaxation of a MINLP model for the problem and an interior point algorithm able to attain near-optimal solutions. Zhao et al. (2017) also focused on a problem in steel production, but they also considered the resulting detrimental effects on the environment of the byproduct gases. Zhao et al. proposed a MILP formulation for the problem whose objective is to minimize the sum of the electricity purchasing costs and a penalty expressed in terms of the different levels of byproduct gases emitted during the time horizon considered for production. The authors achieved a 29.7% reduction in electricity costs while reducing the amount of volume of byproduct gases, by applying their MILP formulation to a real-world case study.

Zeng et al. (2021) considered the problem of optimizing the makespan and the TEC of the production of tissue paper mills, which consists of a two-stage flexible flow shop. The first stage prepares the base paper for the final refinement and packaging performed by the second stage. Machines' setup costs and transportation costs of materials between the first and the second stage are considered together with the usual processing costs. As a solution approach, the authors proposed a hybrid evolutionary algorithm, which they improved by adding a VNS as an enhancing local search. The authors showed the effectiveness of their solution on the case study of a tissue paper mill located in China.

Forghani et al. (2021) addressed the production scheduling of slurry ball mills in the tile industry under preventive maintenance. The objective of the production is the minimization of a linear combination of the TEC and a measure of the deviation of the actual schedule from the ideal schedule in terms of maintenance requirements. The authors proposed a solution algorithm that first determines maintenance operations and then minimizes the TEC by means of MILP. The experimental tests show promising results with a small case study, which amount to a 73% reduction in peak load and a 26% reduction in energy costs, on average, with respect to an energy-oblivious production model. These preliminary results suggest that managers may avoid the considerable expense of equipment upgrade (to reduce maintenance time and downtimes), if they are able to properly model their manufacturing problem within an energy-conscious framework, e.g., as a version of the BJSTP.

Wang et al. (2016) considered a version of the BJSTP that is relevant for the glass-ceramic industry. Specifically, Wang et al. focused on the problem of scheduling glass ceramization on a furnace that processes pallets of glass materials in batches with the objective of minimizing both the makespan and the TEC. The authors modeled such a scenario as a single-machine batch

Table 6
A compendium of real-life applications of versions of the BJSTP.

Application	Literature
Iron-steel plants	Cao et al. (2021)
Rolling sector	Tan et al. (2017); Zhao et al. (2018)
Steel-making continuous casting	Guirong & Qiqiang (2017); Pan et al. (2019, 2022)
Environment-aware steel production	Yang et al. (2018); Zhao et al. (2017)
Mills	Forghani et al. (2021); Zeng et al. (2021)
Glass manufacturing	Wang et al. (2016)
Rubber tires	Guo et al. (2022)

scheduling version of the BJSTP, where the TEC is expressed in terms of the different operational temperatures of the furnace. Similarly to Wang et al. (2018b) and Anghinolfi et al. (2021) (see Section 3), Wang et al. used an exact algorithm based on the ϵ-constraint paradigm to compute the optimal Pareto front. In addition, the authors proposed two heuristics based on decomposition to deal with large-scale instances of the problem. In particular, the first heuristic deals with batch construction, while the second one performs batch sequencing and determines the operational temperatures of the furnace. The authors successfully applied their heuristic to a manufacturing site in Shanghai, China, for which they obtained up to 16.8% energy cost reduction, at the expense of a 20.8% deterioration in makespan.

Finally, Guo et al. (2022) modeled rubber tire production as a flexible job shop problem whose objective is the minimization of the total production and energy costs. In particular, the authors focused on the banburying process, which converts raw rubber into a strong material with elastic properties. The authors proposed a hybrid approach based on a GA and Petri nets. The nodes in the Petri net model different states for energy usage, and simulation provides a measure of the operational time. The genetic component of the solution approach handles the allocation of the resources needed to complete the production plan.

Table 6 summarizes the different applications of TOU scheduling discussed in this subsection.

5. Conclusions

In this article, we reviewed the most prominent literature on energy-efficient job scheduling with TOU-based tariffs, whose theoretical and practical relevance has considerably increased in the last few years. This specific class of scheduling problems is characterized by a pricing scheme induced by TOU-based tariffs, that drive customers' demand to relieve the environmental impact of peak energy generation. We have presented the motivation behind the worldwide strive for sustainability and classified the flourishing literature from a historical point of view while dealing with the computational details of the involved solution approaches. Our effort focused on providing researchers and practitioners with a comprehensive view of the literature that may guide them toward the most important theoretical advances and practical applications in sustainable manufacturing. Toward this end, we proposed a taxonomy that encompasses and categorizes all the most significant research outputs. The presentation of the taxonomy closely follows the development of the literature, to provide a complete and consistent framework for job scheduling with TOU-based tariffs as a whole.

As concerns further developments in the field, we believe necessary to devote further efforts to the study of the combinatorial and polyhedral properties of the considered problems. Despite the existence of many significant results for single-machine problems, scheduling on multiple machines still mainly relies on heuristic or metaheuristic algorithms. Extending the knowledge on the combinatorics of such problems would provide assistance in speeding up the existing heuristics, and uncovering characterizing polyhedral properties would possibly enable the development of faster approximation or exact algorithms. Such advancements would be essential to tackle many application cases, where the involved real-world scenarios require the application of efficient and scalable algorithms. Finally, future research should also be focused on investigating the limitations of TOU-based tariffs in containing the environmental impact of scheduling and production as well as possible alternative pricing schemes that may prove competitive or even more effective.

Acknowledgment

The authors are grateful to the reviewers for their careful reading of the article as well as their valuable comments. The first author acknowledges support from the Belgian National Fund for Scientific Research (FNRS) via the grant FNRS PDR 40007831, and the Fondation Louvain via the grant "COALESCENS".

References

Aghelinejad, M., Ouazene, Y., & Yalaoui, A. (2019a). Complexity analysis of energy–efficient single machine scheduling problems. *Operations Research Perspectives, 6*, 100105.

Aghelinejad, M. M., Masmoudi, O., Ouazene, Y., & Yalaoui, A. (2020a). Multi-state two-machine permutation flow shop scheduling optimisation with time-dependent energy costs. *IFAC-PapersOnLine, 53*(2), 11156–11161.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2016). Machine and production scheduling under electricity time varying prices. In *2016 IEEE international conference on industrial engineering and engineering management (IEEM)* (pp. 992–996). IEEE.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2017). Preemptive scheduling of a single machine with finite states to minimize energy costs. In A. Sforza, & C. Sterle (Eds.), *Optimization and decision science: Methodologies and applications: vol. 217* (pp. 591–599). Springer International Publishing.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2018a). Energy efficient scheduling problems under time-of-use tariffs with different energy consumption of the jobs. *IFAC-PapersOnLine, 51*(11), 1053–1058.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2018b). Production scheduling optimisation with machine state and time-dependent energy costs. *International Journal of Production Research, 56*(16), 5558–5575.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2019b). Single-machine scheduling problem with total energy consumption costs minimization. *IFAC-PapersOnLine, 52*(13), 409–414.

Aghelinejad, M. M., Ouazene, Y., & Yalaoui, A. (2020b). Energy-cost-aware flow-shop scheduling systems with state-dependent energy consumptions. *IOP Conference Series: Earth and Environmental Science, 463*, 012163.

Alvarez-Meaza, I., Zarrabeitia-Bilbao, E., Rio-Belver, R.-M., & Garechana-Anacabe, G. (2021). Green scheduling to achieve green manufacturing: Pursuing a research agenda by mapping science. *Technology in Society, 67*, 101758.

Anghinolfi, D., Paolucci, M., & Ronco, R. (2021). A bi-objective heuristic approach for green identical parallel machine scheduling. *European Journal of Operational Research, 289*, 416–434.

Badri, S. A., Daghbandan, A., Aghabeiginiyay Fatalaki, Z., & Mirzazadeh, M. (2021). Flow shop scheduling under time-of-use electricity tariffs using fuzzy multi-objective linear programming approach. *Journal of Mathematical Modeling, 9*(2), 215–227.

Balas, E. (2018). *Disjunctive programming* (1st ed.). Springer Publishing Company, Incorporated.

Brenner, U. (2008). A faster polynomial algorithm for the unbalanced hitchcock transportation problem. *Operations Research Letters, 36*(4), 408–413.

Cai, W., Wang, L., Li, L., Xie, J., Jia, S., Zhang, X., Jiang, Z., & Lai, K.-H. (2022). A review on methods of energy performance improvement towards sustainable

manufacturing from perspectives of energy monitoring, evaluation, optimization and benchmarking. *Renewable and Sustainable Energy Reviews, 159*, 112227.
Cao, J., Pan, R., Xia, X., Shao, X., & Wang, X. (2021). An efficient scheduling approach for an iron-steel plant equipped with self-generation equipment under time-of-use electricity tariffs. *Swarm and Evolutionary Computation, 60*, 100764.
Chawla, S., Devanur, N. R., Holroyd, A. E., Karlin, A. R., Martin, J. B., & Sivan, B. (2017). Stability of service under time-of-use pricing. In *Proceedings of the 49th annual ACM sigact symposium on theory of computing, STOC 2017* (pp. 184–197). New York, NY, USA: Association for Computing Machinery.
Che, A., Zeng, Y., & Lyu, K. (2016). An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs. *Journal of Cleaner Production, 129*, 565–577.
Che, A., Zhang, S., & Wu, X. (2017). Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. *Journal of Cleaner Production, 156*, 688–697.
Chen, B., & Zhang, X. (2019). Scheduling with time-of-use costs. *European Journal of Operational Research, 274*(3), 900–908.
Chen, L., Megow, N., Rischke, R., Stougie, L., & Verschae, J. (2021). Optimal algorithms for scheduling under time-of-use tariffs. *Annals of Operations Research, 304*, 85–107.
Chen, S.-H., Liou, Y.-C., Chen, Y.-H., & Wang, K.-C. (2019). Order acceptance and scheduling problem with carbon emission reduction and electricity tariffs on a single machine. *Sustainability, 11*(19), 5432.
Chen, W., Wang, J., Yu, G., & Hu, Y. (2022). Energy-efficient hybrid flow-shop scheduling under time-of-use and ladder electricity tariffs. *Applied Sciences, 12*(13), 6456.
Cheng, J., Chu, F., Chu, C., & Xia, W. (2016a). Bi-objective optimization of single-machine batch scheduling under time-of-use electricity prices. *RAIRO - Operations Research, 50*(4), 715–732.
Cheng, J., Chu, F., Ming Liu, & Xia, W. (2016b). Single-machine batch scheduling under time-of-use tariffs: New mixed-integer programming approaches. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 003498–003503). IEEE.
Cheng, J., Chu, F., Xia, W., Ding, J., & Ling, X. (2014). Bi-objective optimization for single-machine batch scheduling considering energy cost. In *2014 international conference on control, decision and information technologies (CoDIT)* (pp. 236–241). IEEE.
Cheng, J., Chu, F., & Zhou, M. (2018). An improved model for parallel machine scheduling under time-of-use electricity price. *IEEE Transactions on Automation Science and Engineering, 15*(2), 896–899.
Cheng, J., Wu, P., & Chu, F. (2019). Mixed-integer programming for unrelated parallel machines scheduling problem considering electricity cost and makespan penalty cost. In *2019 international conference on industrial engineering and systems management (IESM)* (pp. 1–5). IEEE.
Cheng, X., Gao, F., Yan, C., Guan, X., Liu, K., Chen, S., … Cai, J. (2016c). Permutation flow shop scheduling with delay time under time-of-use electricity tariffs. In *2016 12th world congress on intelligent control and automation (WCICA)* (pp. 2743–2748). IEEE.
Choudhury, T. T., Paul, S. K., Rahman, H. F., Jia, Z., & Shukla, N. (2020). A systematic literature review on the service supply chain: Research agenda and future research directions. *Production Planning and Control, 31*(16), 1363–1384.
Cui, W., & Lu, B. (2021). Energy-aware operations management for flow shops under TOU electricity tariff. *Computers and Industrial Engineering, 151*, 106942.
Cui, W., Sun, H., & Xia, B. (2020). Integrating production scheduling, maintenance planning and energy controlling for the sustainable manufacturing systems under TOU tariff. *Journal of the Operational Research Society, 71*(11), 1760–1779.
Deng, L.-Y. (2006). The cross-entropy method: A unified approach to combinatorial optimization, Monte–Carlo simulation, and machine learning. *Technometrics, 48*(1), 147–148.
Derigs, U. (1988). *The hitchcock transportation problem* (pp. 172–182)). Berlin, Heidelberg: Springer Berlin Heidelberg.
Despeisse, M., Ball, P., Evans, S., & Levers, A. (2012). Industrial ecology at factory level – a conceptual model. *Journal of Cleaner Production, 31*, 30–39.
Ding, J., Schulz, S., Shen, L., Buscher, U., & Lü, Z. (2021). Energy aware scheduling in flexible flow shops with hybrid particle swarm optimization. *Computers and Operations Research, 125*, 105088.
Ding, J.-Y., Song, S., Zhang, R., Chiong, R., & 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*(2), 1138–1154.
Dobbs, I. M. (2015). Ladder pricing – a new form of wholesale price discrimination. *International Journal of the Economics of Business, 22*(1), 1–21.
Energy Information Administration (EIA), U. (2021). Manufacturing energy consumption survey,. https://www.eia.gov/consumption/manufacturing/pdf/MECS%202018%20Results%20Flipbook.pdf.
Fang, K., Uhan, N. A., Zhao, F., & Sutherland, J. W. (2016). Scheduling on a single machine under time-of-use electricity tariffs. *Annals of Operations Research, 238*(1), 199–227.
Forghani, A., Lotfi, M., Ranjbar, M., & Sadegheih, A. (2021). Hierarchical framework for maintenance and production scheduling of continuous ball mills in tile industries under TOU electricity pricing. *Journal of Cleaner Production, 327*, 129440.
Gagniuc, P. A. (2017). *Markov Chains: From Theory to Implementation and Experimentation (1st ed)*. John Wiley & Sons, Ltd.
Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research, 248*, 744–757.
Gao, K., Huang, Y., Sadollah, A., & Wang, L. (2020). A review of energy-efficient scheduling in intelligent production systems. *Complex and Intelligent Systems, 6*, 237–249.
Garey, M. R., & Johnson, D. S. (1990). *Computers and intractability; a guide to the theory of NP-completeness*. USA: W. H. Freeman & Co..
Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of metaheuristics* (2nd ed.). Springer Publishing Company, Incorporated.
Giret, A., Trentesaux, D., & Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. *Journal of Manufacturing Systems, 37*, 126–140.
Gong, X., De Pessemier, T., Joseph, W., & Martens, L. (2016). A generic method for energy-efficient and energy-cost-effective production at the unit process level. *Journal of Cleaner Production, 113*, 508–522.
Gong, X., De Pessemier, T., Martens, L., & Joseph, W. (2019). Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation. *Journal of Cleaner Production, 209*, 1078–1094.
Graham, R., Lawler, E., Lenstra, J., & Kan, A. (1977). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics, 5*, 287–326.
Guirong, W., & Qiqiang, L. (2017). Solving the steelmaking-continuous casting production scheduling problem with uncertain processing time under the TOU electricity price. In *2017 Chinese automation congress (CAC)* (pp. 1383–1388). IEEE.
Guo, J., Luo, Q., Liang, P., & Ouyang, J. (2022). A GAPN approach for the flexible job-shop scheduling problem with indirect energy and time-of-use electricity pricing. *Processes, 10*(5), 832.
Hadera, H., Harjunkoski, I., Sand, G., Grossmann, I. E., & Engell, S. (2015). Optimization of steel production scheduling with complex time-sensitive electricity cost. *Computers and Chemical Engineering, 76*, 117–136.
Haimes, Y., Lasdon, L., & Wismer, D. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics, 1*, 296–297.
Ho, M. H., Hnaien, F., & Dugardin, F. (2021). Electricity cost minimisation for optimal makespan solution in flow shop scheduling under time-of-use tariffs. *International Journal of Production Research, 59*(4), 1041–1067.
Ho, M. H., Hnaien, F., & 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*, 105788.
Hu, F., Feng, X., & Cao, H. (2018). A short-term decision model for electricity retailers: Electricity procurement and time-of-use pricing. *Energies, 11*, 3258.
International Energy Agency (2016). World Energy Outlook 2016, https://www.eia.gov/outlooks/ieo/pdf/0484(2016).pdf.
International Energy Agency (2019). World Energy Outlook 2019, https://iea.blob.core.windows.net/assets/98909c1b-aabc-4797-9926-35307b418cdb/WEO2019-free.pdf.
Jiang, E.-d., & Wang, L. (2020). Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity prices. *Knowledge-Based Systems, 204*, 106177.
Kannan, D., Khodaverdi, R., Olfat, L., Jafarian, A., & Diabat, A. (2013). Integrated fuzzy multi criteria decision making method and multi-objective programming approach for supplier selection and order allocation in a green supply chain. *Journal of Cleaner Production, 47*, 355–367.
Karatzas, A., Ancarani, A., Fratocchi, L., Di Stefano, C., & Godsell, J. (2022). When does the manufacturing reshoring strategy create value? *Journal of Purchasing and Supply Management, 28*(3), 100771.
Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - international conference on neural networks: vol. 4* (pp. 1942–1948).
Kim, M., Hiroyasu, T., Miki, M., … Watanabe, S. (2004). SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, … H.-P. Schwefel (Eds.), *Parallel problem solving from nature - PPSN VIII* (pp. 742–751). Berlin, Heidelberg: Springer Berlin Heidelberg.
Kong, M., Xu, J., Zhang, T., Lu, S., Fang, C., & Mladenovic, N. (2021). Energy-efficient rescheduling with time-of-use energy cost: Application of variable neighborhood search algorithm. *Computers and Industrial Engineering, 156*, 107286.
Kurniawan, B., Chandramitasari, W., Gozali, A. A., Weng, W., & Fujimura, S. (2020a). Triple-chromosome genetic algorithm for unrelated parallel machine scheduling under time-of-use tariffs. *IEEJ Transactions on Electrical and Electronic Engineering, 15*(2), 208–217.
Kurniawan, B., & Fujimura, S. (2020). Optimization of bi-objective permutation flow shop scheduling with electricity cost consideration. *IOP Conference Series: Materials Science and Engineering, 909*, 012045.
Kurniawan, B., Gozali, A. A., Weng, W., & Fujimura, S. (2017). A genetic algorithm for unrelated parallel machine scheduling minimizing makespan cost and electricity cost under time-of-use (TOU) tariffs with job delay mechanism. In *2017 IEEE international conference on industrial engineering and engineering management (IEEM)* (pp. 583–587). Singapore: IEEE.
Kurniawan, B., Gozali, A. A., Weng, W., & Fujimura, S. (2018). A mix integer programming model for bi-objective single machine with total weighted tardiness and electricity cost under time-of-use tariffs. In *2018 IEEE international conference on industrial engineering and engineering management (IEEM)* (pp. 137–141). IEEE.
Kurniawan, B., Song, W., Weng, W., & Fujimura, S. (2020b). Distributed-elite local

search based on a genetic algorithm for bi-objective job-shop scheduling under time-of-use tariffs. *Evolutionary Intelligence, 14*, 1581–1595.
Lambert, L. A., Tayah, J., Lee-Schmid, C., Abdalla, M., Abdallah, I., Ali, A. H. M., Esmail, S., & Ahmed, W. (2022). The EU's natural gas cold war and diversification challenges. *Energy Strategy Reviews, 43*, 100934.
Lee, S., Do Chung, B., Jeon, H. W., & Chang, J. (2017). A dynamic control approach for energy-efficient production scheduling on a single machine under time-varying electricity pricing. *Journal of Cleaner Production, 165*, 552–563.
Li, M., Li, T., Peng, S., & Guo, Y. (2018). Batch scheduling of remanufacturing flexible job shop for minimal electricity- and time-cost. In S. Wang, M. Price, M. K. Lim, Y. Jin, Y. Luo, & R. Chen (Eds.), *Recent advances in intelligent manufacturing*. In *Communications in computer and information science: vol. 923* (pp. 300–307). Springer Singapore.
Luo, H., Du, B., Huang, G. Q., Chen, H., & Li, X. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics, 146*(2), 423–439.
Moon, J.-Y., Shin, K., & Park, J. (2013). Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *The International Journal of Advanced Manufacturing Technology, 68*(1), 523–535.
Nikzad, M., & Samimi, A. (2020). Integration of optimal time-of-use pricing in stochastic programming for energy and reserve management in smart micro–grids. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 44*, 1449–1466.
Pan, R., Li, Z., Cao, J., Zhang, H., & Xia, X. (2019). Electrical load tracking scheduling of steel plants under time-of-use tariffs. *Computers and Industrial Engineering, 137*, 106049.
Pan, R., Wang, Q., Li, Z., Cao, J., & Zhang, Y. (2022). Steelmaking-continuous casting scheduling problem with multi-position refining furnaces under time-of-use tariffs. *Annals of Operations Research, 310*(1), 119–151.
Para, J., Del Ser, J., & Nebro, A. J. (2022). Energy-aware multi-objective job shop scheduling optimization with metaheuristics in manufacturing industries: A critical survey, results, and perspectives. *Applied Sciences, 12*(3), 1491.
Park, M.-J., & Ham, A. (2022). Energy-aware flexible job shop scheduling under time-of-use pricing. *International Journal of Production Economics, 248*, 108507.
Pei, Z., Wan, M., Jiang, Z.-Z., Wang, Z., & Dai, X. (2021). An approximation algorithm for unrelated parallel machine scheduling under TOU electricity tariffs. *IEEE Transactions on Automation Science and Engineering, 18*(2), 743–756.
Peng, L., Yang, M., & Xiao, R. (2021). An integer programming model for flow shop scheduling under TOU and tiered electricity price. *IOP Conference Series: Earth and Environmental Science, 692*(2), 022105.
Penn, M., & Raviv, T. (2021). Complexity and algorithms for min cost and max profit scheduling under time-of-use electricity tariffs. *Journal of Scheduling, 24*(1), 83–102.
Pereira, P., Bašić, F., Bogunovic, I., & Barcelo, D. (2022). Russian-ukrainian war impacts the total environment. *Science of the Total Environment, 837*, 155865.
Pinedo, M. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). Berlin, Germany: Springer.
Qian, S.-y., Jia, Z.-h., & Li, K. (2020). A multi-objective evolutionary algorithm based on adaptive clustering for energy-aware batch scheduling problem. *Future Generation Computer Systems, 113*, 441–453.
Raza, S. A., & Hameed, A. Z. (2021). Models for maintenance planning and scheduling – a citation-based literature review and content analysis. *Journal of Quality in Maintenance Engineering, 28*(4), 873–914.
Rego, M. F., Pinto, J. C. E. M., Cota, L. P., & Souza, M. J. F. (2022). A mathematical formulation and an NSGA-II algorithm for minimizing the makespan and energy cost under time-of-use electricity price in an unrelated parallel machine scheduling. *Peer J Computer Science, 8*, e844.
Renna, P., & Materi, S. (2021). A literature review of energy efficiency and sustainability in manufacturing systems. *Applied Sciences, 11*(16), 7366.
Rocholl, J., Mönch, L., & Fowler, J. (2020). Bi-criteria parallel batch machine scheduling to minimize total weighted tardiness and electricity cost. *Journal of Business Economics, 90*(9), 1345–1381.
Ronco, R. (2022). *Exact and heuristic algorithms for energy-efficient scheduling*. University of Genoa, Italy Ph.D. thesis. http://hdl.handle.net/11567/1068456.
Rubaiee, S., & Yildirim, M. B. (2019). An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling. *Computers and Industrial Engineering, 127*, 240–252.
Rubinstein, R. Y. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology And Computing In Applied Probability, 1*, 127–190.
Saberi-Aliabad, H., Reisi-Nafchi, M., & Moslehi, G. (2020). Energy-efficient scheduling in an unrelated parallel-machine environment under time-of-use electricity tariffs. *Journal of Cleaner Production, 249*, 119393.
Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production, 67*, 197–207.
Sin, I. H., & Chung, B. D. (2020). Bi-objective optimization approach for energy aware scheduling considering electricity cost and preventive maintenance using genetic algorithm. *Journal of Cleaner Production, 244*, 118869.
Soares, I., Alves, M. J., & Henggeler Antunes, C. (2021). A deterministic bounding procedure for the global optimization of a bi-level mixed-integer problem. *European Journal of Operational Research, 291*(1), 52–66.
Tan, M., Yang, H.-l., Duan, B., Su, Y.-x., & He, F. (2017). Optimizing production scheduling of steel plate hot rolling for economic load dispatch under time-of-use electricity pricing. *Mathematical Problems in Engineering, 2017*, 1–13.
Tan, M., Yang, H.-L., & Su, Y.-X. (2019). Genetic algorithms with greedy strategy for green batch scheduling on non-identical parallel machines. *Memetic Computing, 11*(4), 439–452.
Terbrack, H., Claus, T., & Herrmann, F. (2021). Energy-oriented production planning in industry: A systematic literature review and classification scheme. *Sustainability, 13*, 13317.
Trevino-Martinez, S., Sawhney, R., & Shylo, O. (2022). Energy-carbon footprint optimization in sequence-dependent production scheduling. *Applied Energy, 315*, 118949.
Wan, G., & Qi, X. (2010). Scheduling with variable time slot costs. *Naval Research Logistics (NRL), 57*(2), 159–171.
Wang, F., Qin, P., & You, J. (2020a). Discrete optimization model for permutation flow shop under time-of-use electricity tariffs. *Journal of Physics: Conference Series, 1633*, 012146.
Wang, K. (2020). Calibration scheduling with time slot cost. *Theoretical Computer Science, 821*, 1–14.
Wang, S., Liu, M., Chu, F., & Chu, C. (2016). Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *Journal of Cleaner Production, 137*, 1205–1215.
Wang, S., Wang, X., Chu, F., & Yu, J. (2020b). An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. *International Journal of Production Research, 58*(8), 2283–2314.
Wang, S., Wang, X., Yu, J., Ma, S., & Liu, M. (2018a). Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan. *Journal of Cleaner Production, 193*, 424–440.
Wang, S., Zhu, Z., Fang, K., Chu, F., & Chu, C. (2018b). Scheduling on a two-machine permutation flow shop under time-of-use electricity tariffs. *International Journal of Production Research, 56*(9), 3173–3187.
Wu, P., Cheng, J., & Chu, F. (2021). Large-scale energy-conscious bi-objective single-machine batch scheduling under time-of-use electricity tariffs via effective iterative heuristics. *Annals of Operations Research, 296*(1), 471–494.
Yang, Y., Chen, W., Wei, L., & Chen, X. (2018). Robust optimization for integrated scrap steel charge considering uncertain metal elements concentrations and production scheduling under time-of-use electricity tariff. *Journal of Cleaner Production, 176*, 800–812.
Yusuf, L. A., Popoola, K., & Musa, H. (2021). A review of energy consumption and minimisation strategies of machine tools in manufacturing process. *International Journal of Sustainable Engineering, 14*(6), 1826–1842.
Zeng, Z., Chen, X., & Wang, K. (2021). Energy saving for tissue paper mills by energy-efficiency scheduling under time-of-use electricity tariffs. *Processes, 9*(2), 274.
Zhang, H., Fang, Y., Pan, R., & Ge, C. (2018a). A new greedy insertion heuristic algorithm with a multi-stage filtering mechanism for energy-efficient single machine scheduling problems. *Algorithms, 11*(2), 18.
Zhang, H., Wu, Y., Pan, R., & Xu, G. (2021). Two-stage parallel speed-scaling machine scheduling under time-of-use tariffs. *Journal of Intelligent Manufacturing, 32*(1), 91–112.
Zhang, H., Zhao, F., Fang, K., & Sutherland, J. W. (2014). Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Annals, 63*(1), 37–40.
Zhang, M., Yan, J., Zhang, Y., & Yan, S. (2019). Optimization for energy-efficient flexible flow shop scheduling under time of use electricity tariffs. *Procedia CIRP, 80*, 251–256.
Zhang, S., Che, A., Wu, X., & Chu, C. (2018b). Improved mixed-integer linear programming model and heuristics for bi-objective single-machine batch scheduling with energy cost consideration. *Engineering Optimization, 50*(8), 1380–1394.
Zhao, S., Grossmann, I. E., & Tang, L. (2018). Integrated scheduling of rolling sector in steel production with consideration of energy consumption under time-of-use electricity prices. *Computers and Chemical Engineering, 111*, 55–65.
Zhao, X., Bai, H., Shi, Q., Lu, X., & Zhang, Z. (2017). Optimal scheduling of a byproduct gas system in a steel plant considering time-of-use electricity pricing. *Applied Energy, 195*, 100–113.
Zhou, S., Jin, M., & Du, N. (2020). Energy-efficient scheduling of a single batch processing machine with dynamic job arrival times. *Energy, 209*, 118420.
Zhou, S., Li, X., Du, N., Pang, Y., & Chen, H. (2018). A multi-objective differential evolution algorithm for parallel batch processing machine scheduling considering electricity consumption cost. *Computers and Operations Research, 96*, 55–68.