Energy-Efficient Job Shop Scheduling Under Time-Of-Use Pricing

Journal:	*Engineering Optimization*
Manuscript ID	GENO-2021-0055
Manuscript Type:	Research Article
Date Submitted by the Author:	18-Jan-2021
Complete List of Authors:	Ham, Andy; North Carolina Agricultural and Technical State University College of Engineering, Applied Engineering Technology
Keywords:	job shop scheduling, energy-efficient, time-of-use, mixed integer programming, constraint programming

Energy-Efficient Job Shop Scheduling Under Time-Of-Use Pricing

Andy Ham
Applied Engineering Technology, North Carolina A&T State University, Greensboro, NC, USA

***Abstract*—This study addresses energy-efficient job shop scheduling problem under time-of-use pricing. The prices of electricity vary over times, offering factories with a cost saving opportunity. In particular, the proposed energy-aware scheduling model (1) shifts the inevitable idle times to peak-periods since a machine consumes a minimum level of standby energy during the idle time and (2) shifts production operations to off-peak-periods, thus reducing a total energy-cost while completing a production at the same time with the conventional method. A novel constraint programming is proposed as well as two mixed integer programming models based on time-indexed and disjunctive formulations. The computational study demonstrates that the proposed models can save energy-cost without compromising productivity.**

***Index Terms*—energy-efficient; job shop scheduling, time-of-use; mixed integer programming; constraint programming.**

***Note to Practitioners*—The energy-aware scheduling hasn't been embraced by capital-intensive manufacturing industries where energy cost is not even comparable to the revenue the manufacturers yield, so their sole goal is to maximize productivity. However, if we can maintain the same productivity, while significantly reducing the energy cost, the energy-aware production scheduling will be adopted by the industries. This paper provides benchmarking instances and CPLEX source codes, in order to promote related-research, thus expediting the adoption of energy-aware scheduling in manufacturing facilities, which contributes to a stabilization of power-grid in return.**

I. Introduction

New York City buildings will be required to display their rating on energy efficiency starting in 2020. Mid-large sized buildings will not only have to report how energy-efficient they are, but they will also be required to post signs with the letter grades "in a conspicuous location near each public entrance," according to the law (NY Times, 2019).

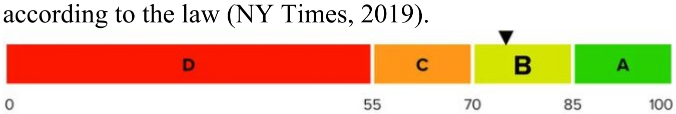


Fig. 1. A preliminary design for the grades, which will be posted at the entrances to thousands of buildings.

A similar rating system could be introduced to manufacturing facilities to gauge an amount of electricity-cost-saving from the advanced production scheduling system that considers time-of-use (TOU) electricity-pricing.

Electricity cannot be efficiently stored so it must be generated, transmitted and consumed simultaneously. In addition, electricity demand is imbalanced over time, which causes a difficulty for electricity suppliers to regulate peak load (Mitra et al., 2012). Fig. 2 illustrates the volatility of hourly demand for three different one-week periods in New York. As a result, electricity price varies, according to the dynamic market demand over time. There are three types of time dependent electricity pricing: time-of-use (TOU) pricing, real-time pricing (RTP) and critical peak pricing (CPP) (Sharma et al. 2015). In the TOU pricing (the most popular type), the electricity price schedule is predefined, but it may vary by hour. This pricing policy offers consumers the opportunity to save costs by shifting their energy consumption from on-peak to off-peak periods. By considering TOU pricing, factories can contribute to stabilizing the power grid during the peak-periods as well as reducing their own cost.

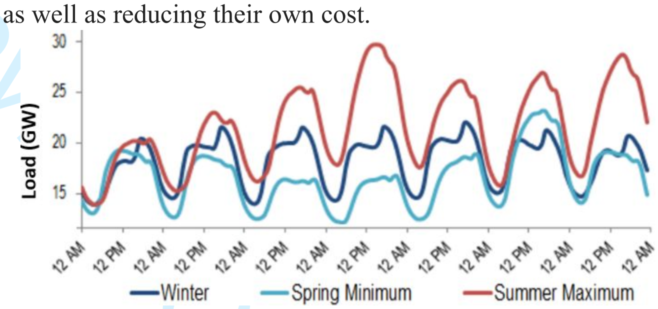


Fig. 2. Historic hourly load pattern for New York Power (Source: energy.gov)

The energy-aware scheduling hasn't been embraced by capital-intensive manufacturing industries such as automobile, display, and semiconductor, where the energy cost is not even comparable to the revenue the manufacturers yield, so their sole goal is to maximize productivity. However, if we can maintain the same productivity, while significantly reducing the energy cost, energy-aware production scheduling will be adopted by the industries. In fact, non-capital-intensive industries may not be even able to afford the advanced scheduling system which could cost a half million US dollars (servers, software licensing, development, etc.) plus salary for supporting workforce.

This paper demonstrates that we can reduce electricity-costs without compromising productivity by studying a job shop scheduling problem that has wide applications in manufacturing facilities, in the context of TOU pricing.

II. Literature review

A. Energy-Efficient Scheduling Under Time-Of-Use Pricing

Moon et al. (2013) addressed the unrelated parallel machine scheduling problem under TOU pricing, after noticing that manufacturing industries pay different electricity charges depending on the time of day (i.e., peak-load, mid-load, and off-peak-load). Similarly, Wang and Li (2013) found that previous research on the applications of TOU and other electricity demand response programs have been limited to residential and commercial buildings, while largely neglecting industrial manufacturing systems.

Table I summarizes the articles related to production scheduling under TOU pricing and contrasts with this paper. Column 3 records the number of machines being tested in each article. When there are multiple stages, the column first records the number of machines followed by the number of stages. The table shows that single stage environment have been extensively studied, while job shop environment have not been addressed, to the best of our knowledge.

B. Job Shop Scheduling Problem

The classical job shop scheduling problem (JSP) schedules a set of jobs on a set of machines with the objective to minimize a maximum completion time over all jobs, subjected to the constraint that each job has an ordered set of operations, each of which must be processed on a predefined machine. This problem has been one of the best-known combinatorial optimization problems since Graham (1966) firstly introduced it.

There are three widely used mixed integer programming (MIP) formulations for JSP according to main binary decision variable: the time-indexed (Bowman, 1959; Kondili et al. 1988), the rank-based (Wagner, 1959), and the disjunctive (Manne, 1960) formulation. This paper extends the time-indexed and disjunctive formulations to capture TOU pricing for the first time.

C. Constraint programming

Google OR-Tools and IBM CP Optimizer are major players in the constraint programming (CP) solver market. OR-Tools is an open-source solver developed by Google and CP Optimizer (abbreviated CPO) is an IBM proprietary solver targeted towards industrial scheduling problems. Col and Teppan (2019) compared the performance of two solvers on JSP and found that OR-Tools performed not much worse than CPO on the small-scale instances, but the sharp difference was monitored as the problem size increased. Therefore, this study exploits CPO that has an efficient representation of the interval and sequence variables. CPO utilizes a temporal relaxation technique in a similar way that mathematical programming employs linear relaxation (Laborie and Rogerie, 2016). Large neighborhood search within CPO consists of a process of continual relaxation and re-optimization: a first solution is computed and iteratively improved. Each iteration consists of a relaxation step followed by a reoptimization of the relaxed solution. This process continues until some condition is satisfied, typically, when the solution can be proved to be optimal.

CP has been successfully applied to many production scheduling problems, dominating MIP in most cases. However, CP has been yet adopted for TOU production scheduling problems, although CP seems to be a great fit compared to MIP that relies on lots of binary decision variables to calculate the energy-cost at time of use for the problem under study.

This paper proposes a CP formulation for JSP-TOU for the first time.

III. Problem Description and Solutions

Consider a job shop environment that consists of a set of heterogeneous machines ($k \in K$) and a set of jobs ($i \in I$). Each

TABLE I
Selected Articles Related To Production Scheduling Under TOU Pricing

Year	Environment	Size	Approach	Feature	Publication
Moon et al. (2013)	Single stage	3	GA	—	International Journal of Advanced Manufacturing Technology
Wang & Li (2013)	Flow shop	1/3	PSO	—	Energy
Shrouf et al. (2014)	Single stage	1	MIP/GA	On/Off	Journal of Cleaner Production
Ding et al. (2015)	Single stage	20	MIP/column generation	—	IEEE Transactions on Automation Science and Engineering
Che et al. (2016)	Single stage	1	Insertion heuristic	—	Journal of Cleaner Production
Fang et al. (2016)	Single stage	1	AM	Muiti-speed	Annals of Operations Research
Cheng et al. (2017)	Single stage	1	MIP/heuristic	Batching	Computers & Industrial Engineering
Che et al. (2017)	Single stage	20	MIP/Insertion heuristic	—	Journal of Cleaner Production
Kurniawan et al. (2017)	Single stage	15	MIP/GA	—	IEEE International Conference on IEEM
Zhang et al. (2018)	Single stage	1	Insertion heuristic	—	Algorithms
Zeng et al. (2018)	Single stage	500	MIP/Insertion heuristic	—	Engineering Optimization
Zhang et al. (2018)	Single stage	1	MIP	Batching	Engineering optimization
Rubaiee et al. (2018)	Single stage	1	MIP/GA	—	IEEE Transactions on Engineering Management
Wang et al. (2018)	Flow shop	1/2	MIP/heuristic	—	International Journal of Production Research
Cota et al. (2018)	Single stage	5	MIP/Heuristic	Muiti-speed; Setup	International Transactions in Operational Research
Tan et al. (2018)	Single stage	12	MIP	Batching	Operational Research
Wu et al. (2019)	Single stage	1	Constructive heuristic	Batching	Annals of Operations Research
Tan et al. (2019)	Single stage	20	MIP/GA	Batching	Memetic Computing
This paper	Job shop	15/15	MIP/CP/Combined	—	—

job i consists of a set of operations ($J_i = \{1,\ldots,n_i\}$). Each operation needs to be processed in a predetermined order (known as precedence constraints) for a given job. A machine can perform at most one operation at a time. Each operation must be processed by a predefined machine. In addition to the standard JSP problem, a total of energy cost (TC) is considered in this energy-aware scheduling approach. In particular, the prices of electricity vary over discrete time intervals.

A. Integer Linear Programing Models

Parameters:
I: set of jobs.
n_i: number of operations of job $i \in I$.
J_i: set of operations of job $i \in I$, that is, $J_i = \{1,\ldots,n_i\}$.
O: set of job and operation pairs, that is, $O = \{(i,j) | i \in I, j \in J_i\}$.
O_o: set of operations which can be processed by the same machine with operation $o \in O$.
Q: set of precedence operation pairs, that is, $Q = \{(o,q) | o = (i,j), q = (i,j+1), \forall i \in I, j = 1,\ldots n_i - 1\}$.
K: set of all machines.
k_o: predetermine machine to process $o \in O$.
p_o: processing time of $o \in O$
h : time horizon or an upper bound of makespan ($\sum_{o \in O} p_o$)
T: set of start times $T = \{0,\ldots,h-1\}$.
c_t: price of electricity during time interval $[t,t+1)$, $t \in T$.
$c_{o,t}$: total price of electricity consumed during time interval $[t,t+p_o)$, that is, $c_{o,t} = \sum_{\tau=t}^{t+p_o-1} c_\tau$.
S: the electric power consumed by supporting facilities.
D_k: the electric power consumed at the idle time of machine k.
P_o: electric power consumed when operation o.
Decision variables:
$X_{o,t}$: 1 if $o \in O$ starts at time slot t and 0 otherwise.
$Z_{o,\hat{o},k}$: 1 if $o \in O$ proceeds $\hat{o} \in O$ on machine k and 0 otherwise.
L_t : 1 if $t \in T$ is less than the makespan and 0 otherwise.
eP: total energy cost during production.
eI: total energy cost during idle.
eC: total energy cost when system is on.
$cmax$: makespan

1) MIP-1: Time Indexed Formulation

$$Min\ eP + eI + eC \tag{A1}$$

$$\sum_{t \in T} X_{o,t} = 1 \qquad \forall o \in O \tag{A2}$$

$$\sum_{t \in T} X_{o,t}(t + p_o) \le \sum_{t \in T} X_{q,t} \cdot t \qquad \forall (o,q) \in Q \tag{A3}$$

$$\sum_{o \in O: k_o = k} \sum_{\tau = t - p_o + 1}^{t} X_{o,\tau} \le 1 \qquad \forall k \in k, t \in T \tag{A4}$$

$$L_{t-1} \ge L_t \qquad t \in T \setminus \{0\} \tag{A5}$$

$$L_{t+p_o-1} \ge X_{o,t} \qquad \forall o \in O, t \in T \tag{A6}$$

$$cmax = \sum_{t \in T} L_t \tag{A7}$$

$$eP = \sum_{o \in O} \sum_{t \in T} c_{o,t} p_o X_{o,t} \tag{A8}$$

$$eI = \sum_{k \in K} \sum_{t \in T} c_t D_{k_o} L_t - \sum_{o \in O} \sum_{t \in T} c_{o,t} D_{k_o} X_{o,t} \tag{A9}$$

$$eC = S \sum_{t \in T} c_t L_t \tag{A10}$$

Objective (A1) is the total of energy cost during the production, idle, and system on. Constraint (A2) ensures that each operation starts processing at exactly one time slot. Constraint (A3) is the precedence constraint. It ensures that all operations of a job are executed in the given order. Constraint (A4) enforces each time interval $[t,t+1)$ of each machine can process at most one operation. Constraints (A5)-(A6) link makespan with the starting time of an operation. Constraint (A7) determines the makespan. Constraints (A8)-(A10) calculate the energy cost during production, idle, and system on, respectively. Note that, in (A9), the first term is total idle cost when the system is on and the second term is the idle cost when machine is under production.

2) MIP-2: Disjunctive Formulation

The time-indexed model could reflect the TOU pricing relatively easily since the time-index is directly linked to the time of use pricing. However, the time-indexed model carries lots of binary variables. On the other hand, the disjunctive formulation employs only continuous variables lacking a direct connection to TOU pricing. Therefore, a piecewise linear function of IBM CPLEX is utilized.

Functions:
fs$[o](t)$: In order to make a computation more efficient, we pre-compute a piecewise linear function that gives the energy price of an operation when started at time t, that is the sum of the price during time interval $[t,t+p_o-1)$.
fe(t): we pre-compute a piecewise linear function that gives the price of an operation when ended at time t, that is the sum of the price during time interval $[o,t-1)$.

Decision variables:
Y_o : start time of operation o.

$$Min\ eP + eI + eC \tag{B1}$$

$$Y_o + p_o \le Y_q \qquad \forall o \in Q \tag{B2}$$

$$Y_o \ge Y_{\hat{o}} + p_{\hat{o}} - h \cdot Z_{o,\hat{o},k_o} \qquad \forall (o,\hat{o}) \in O_o \tag{B3}$$

$$Y_{\hat{o}} \ge Y_o + \hat{o} - h \cdot (1 - Z_{o,\hat{o},k_o}) \qquad \forall (o,\hat{o}) \in O_o \tag{B4}$$

$$cmax \ge Y_o + p_o \qquad t \in T \setminus \{0\} \tag{B5}$$

$$eP = \sum_{o \in O} \sum_{t \in T} fs[o](Y_o) \cdot p_o \tag{B6}$$

$$eI = \sum_{k \in K} \sum_{t \in T} fe(L_t) \cdot D_k - \sum_{o \in O} \sum_{t \in T} fs[o](Y_o) \cdot D_{k_o} \tag{B7}$$

$$eC = S \sum_{t \in T} fe(L_t) \tag{B8}$$

Objective (B1) is the total energy cost during the production, idle, and system on. Constraint (B2) is the precedence constraint. It ensures that all operations of a job are executed in the given order. Constraints (B3)-(B4) ensure that no two jobs can be scheduled on the same machine at the same time. Constraint (B5) determines the makespan. Constraints (B6)-(B8) calcualte the energy cost during production, idle, and system on, respectively.

Fig. 4 contrasts the conventional JSP with JSP-TOU by using a small benchmark instance (E11). The upper Gantt-chart (4a) shows the detailed production schedule with an objective of makespan minimization and the lower (4b) demonstrates the schedule with an objective of TC minimization while maintaining the same productivity (makespan). The proposed energy-aware scheduling model (1) shifts the inevitable idle times to peak-period since a machine consumes a minimum standby energy during idle time and (2) shifts production operations to off-peak-period, thus reducing TC by 2.3% while completing the production at the same time with the conventional method.

The heat map on the bottom of the figure indicates the hourly day-ahead TOU pricing/mWh for January 30, 2019 (source: hourlypricing.comed.com). The solid-line boxes represent production operations and dash-lines indicate idle times, marked by associated costs in blue and red colors, respectively. For instance, M1 processes J1 O1 for 8-time units, consuming 3 Mkw at each time unit, resulting in a total cost of 3 × ($26 + $26 + $27 + $26 + $26+ $29 + $36 + $46). M1 experiences an idle interval at 58 for 2-time units, consuming 2 Mkw at each time unit, resulting in a total cost of 2 × ($53 + $47).

B. Constraint Programming Model

The preliminary study showed that both MIP models failed to find any feasible solution for the mid- to large-sized test instances. This study applies CP that is a complementary method to MIP in that CP can also prove optimality.

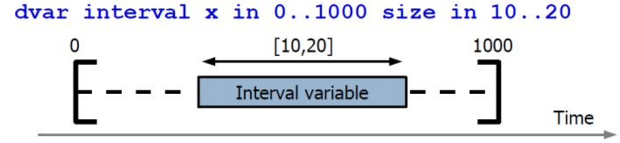


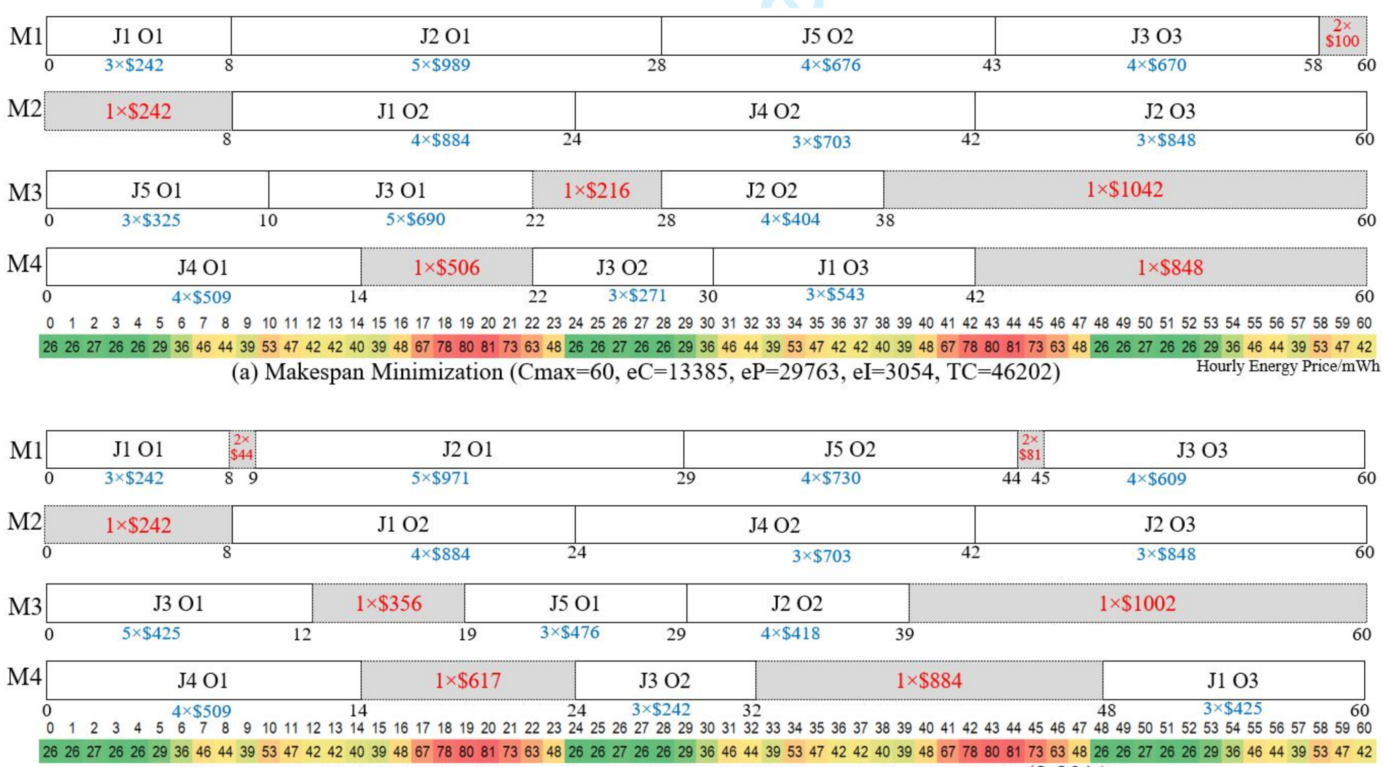
Fig. 3. A notion of interval variable (Laborie et al. 2009).

CPO extends the classical CP paradigm by introducing additional mathematical concepts such as intervals, sequences, and functions (Laborie and Rogerie 2008). Among them, the interval variable (Fig. 3) provides a key notation of the modeling. An ***interval variable*** a is a decision variable whose domain is a subset of $\{\perp\} \cup \{[s, e) : s, e \in \mathbb{Z} \wedge s \leq e\}$. A Boolean *presenceOf(a)* denotes the presence status of the interval variable. It is part of the decisions of the problem to decide whether interval a is present or absent. More precisely, at a solution,

– if a is present, $a = [s, e)$ and *presenceOf (a)* is true. *startOf(a)*, *endOf(a)*, and *lengthOf(a)*, respectively, denote the start time, the end time, and the length of interval variable a.

– else, a is absent, $a = \perp$ and *presenceOf (a)* is false. In our application, an interval variable represents an operation.

On the other hand, a ***sequence variable*** is defined by a set (S) of interval variables. The value of a sequence variable is a permutation of the present intervals in set S. Various constraints on sequence variables are available to state (1) precedence relations between interval variables in the permutation, (2) temporal relations between the intervals (for example, a *noOverlap* constraint states that, according to the permutation order, any interval in the sequence must end before its successor starts). Sequence variables and *noOverlap* constraints are typically used to model disjunctive unary



(a) Makespan Minimization (Cmax=60, eC=13385, eP=29763, eI=3054, TC=46202)

(b) Energy Cost Minimization (Cmax=60, eC=13385, eP=28388, eI=3351, TC=45124 (**2.3%↓**)

Fig. 4. Energy-Aware JSP Under TOU Pricing (E11 instance)

resources and the possible transition constraints between consecutive activities on the resource (like sequence-dependent setup times, costs, traveling distances). In our application, a sequence variable represents a set of operations performed by each machine.

Decision variables:
C: interval representing makespan
T_o: interval representing operation $o \in O$
$Seq_k \leftarrow [T_o]$: collection of variables assigned to machine k.

$$Min\ eP + eI + eC \tag{C1}$$

$$endBeforeStart(T_o, T_q) \quad \forall (o,q) \in Q \tag{C2}$$

$$noOverlap(Seq_k) \quad \forall k \in K \tag{C3}$$

$$startOf(C) = 0 \quad \forall k \in K \tag{C4}$$

$$endOf(C) = Max_{\forall o \in O}\{endOf(T_o)\} \tag{C5}$$

$$eP = \sum_{o \in O} P_o\, startEval(T_o, fs[o]) \tag{C6}$$

$$eI = \sum_{k \in K} endEval(C, fe) D_k - \sum_{o \in O} startEval(T_o, fs[o]) D_{k_o} \tag{C7}$$

$$eC = endEval(C, fe) P_0 \tag{C8}$$

The objective (C1) is to minimize the total energy-cost (TC) which is the total energy consumed for production, idle, and system on. Constraint (C2) is the precedence constraint. It ensures that all operations of a job are executed in the given order. Constraint (C3) prevents intervals in a sequence from overlapping on each machine. Constraints (C4)-(C5) determine the start and end times of makespan. Constraints (C6)-(C8) compute the energy-cost for production, idle, and common, respectively. The numerical expressions (*startEval* and *endEval*) are used to evaluate a piecewise linear function at the start and end of an interval, respectively, for TOU pricing.

In this small test-run, two different objectives (makespan minimization and TC minimization) generated the same makespan of 60 by chance. However, different values of makespan were noticed in large instances since two different objectives lead to different makespan values.

Therefore, a multi-stage optimization approach is adopted in order to demonstrate that the proposed methods can reduce the TC without compromising a productivity. Fig. 5 depicts the proposed approach. At the first stage, the proposed CP model generates a solution that minimizes a makespan. The solution is passed to the second stage where CP searches for a solution that minimizes TC. Namely, when CP starts a search at the second stage, it refers to the solution from the first stage (warm-start).

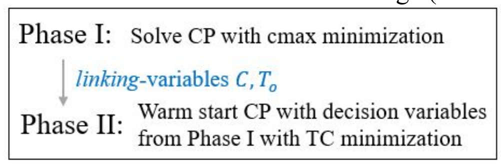


Fig. 5. Two-stage optimization based on warm-start method

IV. Computational Experiments

In this section, the effectiveness of the proposed model is examined. The MIP, CP, and flow control models are all coded in IBM OPL 12.10.0 on a personal computer with an Intel® Core i7-4770 CPU with 16 GB of RAM. All the test instances, results (to construct Gantt), MIP codes, and CP can be found online at: https://github.com/hamcruise/JSPTOU.

A. Problem instances

This author could not obtain the same instances used by the previous researchers in this energy-aware scheduling field, even after many attempts. Using a standard set of instances for a similar problem is a desirable practice, enabling a direct comparison of performance and accuracy among different approaches and researchers. Therefore, this author provides the benchmarking instances, CPLEX MIP, and CP codes, in order to promote related-research, thus expediting the adoption of energy-aware scheduling in manufacturing facilities.

The proposed standard instances were built upon the classical JSP instances by adding energy parameters in a similar method Meng et al. (2019) have been taken. The small JSP benchmark instances proposed by Bilge and Ulusoy (1995) were adopted. The instances have up to 8 jobs and 4 machines. In addition, the following energy-related parameters were added. For all

TABLE II
Comparison of models in terms of energy costs based on Bilge and Ulusoy (1995) Instances

Instance Characteristics			Traditional (makespan minimization)					Energy-Aware Scheduling (TC minimization)									
			CP					CP						MIP-1		MIP-2	
Inst	Size	Cmax	eC	eP	eI	TC	Sec	eC	eP	eI	TC	TC ↓	Sec	TC	Sec	TC	Sec
E11	4/5/3	60	13385	29763	3054	46202	0.03	13385	28388	3351	**45124**	2.3%	0.28	**45124**	1.97	45124	60.00
E21	4/5/3	70	16335	33849	5305	55489	0.03	16335	32313	5778	**54426**	1.9%	4.06	**54426**	15.67	54873	60.00
E31	4/6/4	70	16335	42585	7556	66476	0.04	16335	39013	9347	**64695**	2.7%	0.12	**64695**	2.89	64695	60.00
E41	4/5/5	54	12060	29364	3095	44519	0.02	12060	28791	3288	**44139**	0.9%	0.34	**44139**	4.24	44570	60.00
E51	4/5/3	48	11260	23154	9927	44341	0.02	11260	21600	10584	**43444**	2.0%	0.26	**43444**	1.61	43444	60.00
E61	4/6/3	88	19830	43424	14190	77444	0.03	19830	41320	15160	**76310**	1.5%	0.26	**76310**	23.98	92209	60.00
E71	4/8/3	66	14775	36342	4578	55695	0.02	14775	33056	5649	**53480**	4.0%	0.86	**53480**	4.52	53953	60.00
E81	4/6/4	141	32860	50970	22841	106671	0.03	32860	46363	24578	**103801**	2.7%	0.86	**103801**	44.38	104845	60.00
E91	4/5/4	81	18320	39670	11077	69067	0.02	18320	39502	11203	**69025**	0.1%	0.32	**69025**	8.08	131311	60.00
E101	4/6/4	112	25460	54620	10831	90911	0.25	25460	52718	11169	**89347**	1.7%	0.49	89658	60.00	99242	60.00

TABLE III
COMPARISON OF MODELS IN TERMS OF ENERGY COSTS BASED ON STORER ET AL. (1992) INSTANCES

Instance Characteristics			Traditional (makespan minimization)					Energy-Aware Scheduling (TC minimization)					
Inst	Size	Cmax	eC	eP	eI	TC	Gap	eC	eP	eI	TC	TC ↓	Gap
swv1	10/20/10	1425	333600	1816390	531047	2681037	3.2%	333600	1794564	541614	2669778	0.4%	3.2%
swv2	10/20/10	1481	346610	1866638	549454	2762702	0.4%	346610	1840911	562459	2749980	0.5%	1.0%
swv3	10/20/10	1415	331930	1866791	510465	2709186	4.9%	331930	1846885	521046	2699861	0.3%	4.5%
swv4	10/20/10	1515	355085	1930793	556330	2842208	7.3%	355085	1906351	569846	2831282	0.4%	6.2%
swv5	10/20/10	1456	340740	1882496	482946	2706182	2.9%	340740	1859838	494473	2695051	0.4%	2.8%
swv6	15/20/15	1715	401620	2859281	1241834	4502735	8.3%	401620	2829833	1257211	4488664	0.3%	7.2%
swv7	15/20/15	1651	386745	2758134	1194299	4339178	14.8%	386745	2726525	1210630	4323900	0.4%	12.2%
swv8	15/20/15	1820	426555	2983802	1356291	4766648	11.3%	426555	2949465	1374410	4750430	0.3%	9.4%
swv9	15/20/15	1707	400125	2824922	1262142	4487189	6.6%	400125	2786242	1283842	4470209	0.4%	5.8%
swv10	15/20/15	1826	428140	2975437	1378638	4782215	13.0%	428140	2932266	1402162	4762568	0.4%	10.7%

instances, the common power (S) was set to be 5. The processing powers (P_o) were drawn from the uniform distribution [3,5]. The idle power (D_k) was randomly generated from the set $\{1, 2, 3\}$. The hourly day-ahead TOU pricing/mWh (c_t) was retrieved from hourlypricing.comed.com and duplicated for the rest of the planning horizon.

One of the classic medium to large-sized JSP benchmark instances (Storer et al. 1992) was also adopted to understand a scalability.

B. Results

Table II contrasts the proposed models with the objective of TC minimization for the small benchmark instances. Column 1 identifies the name of the instance, Column 2 identifies the size of instance (*a*/*b*/*c*) in which index *a* denotes number of machines, *b* denotes number of jobs, and *c* denotes the maximum number of operations of each job, and Column 3 records the optimal makespan. Columns 4-18 record energy costs found within 60 seconds. Bold font indicates the optimal. Columns 4-8 record costs and run-time of the CP model with an objective of makespan minimization, while Columns 9-14 contain the results of CP model with an objective of TC minimization. There was 2.0% reduction of TC after applying the proposed models on average. Columns 15-16 (17-19) record the TC and run-time of MIP-1 and MIP-2, respectively. MIP-1 successfully found the optimal solutions within 60 seconds except E101 instance, while MIP-2 could not prove the optimality of any of test instances. Despite of zero-usage of the binary variable of disjunctive formulation (MIP-2), the disjunctive formulation turned out to be ineffective for JSP-TOU. In overall, CP significantly outperformed both MIP models.

Table III contains a similar result for the medium to large-sized benchmark instances. Both MIP models failed to find feasible solutions within the time-limit. Therefore, the table contains results from CP only. Columns 8 and 14 record optimality gaps calculated with 60 seconds run-time since the proposed CP model could not converge to optimal. This additional experimentation identified the potential TC saving by 0.4% without compromising productivity, by considering TOU. However, the optimality gap (6.3% on average) of the model with an objective of TC minimization shows that a further opportunity is left for exploration.

V. CONCLUSION

This study has investigated the energy-efficient job shop scheduling under time-of-use pricing. The prices of electricity vary over times, offering factories with a cost saving opportunity. In particular, the proposed energy-aware scheduling model (1) shifts the inevitable idle times to peak-period since a machine consumes a minimum standby energy during the idle and (2) shifts production operations to off-peak-period, thus reducing a total energy-cost while completing a production at the same time with the conventional method.

A novel constraint programming is proposed as well as two mixed integer programming models based on the time-indexed and disjunctive formulations for the first time. Both MIP-1 and CP proved optimality of small instances. However, MIP could not find feasible solutions of large instances (even with 3 hours run-time), while CP successfully found efficient solutions within a couple of seconds. In overall, CP is a promising method for JSP-TOU, where efficiently calculating energy-price at time of use is crucial.

In future research, it would be interesting to develop heuristics to reduce the significant optimality gap (6.3%) noticed at the large instances. This author provides the benchmarking instances, CPLEX MIP, and CP codes, in order to promote related-research, thus expediting the adoption of energy-aware scheduling in manufacturing facilities, which contributes to a stabilization of power-grid in return.

REFERENCES

[1] Bilge, Umit, and Gunduz Ulusoy. 1995. "A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS." Operations Research 43 (6): 1058–1070.

[2] Bowman, E. H. (1959). The schedule-sequencing problem. Operations Research, 7(5), 621-624.

[3] 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.

[4] 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.

[5] 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.

[6] Cheng, J., Chu, F., Liu, M., Wu, P., & Xia, W. (2017). Bi-criteria single-machine batch scheduling with machine on/off switching under time-of-use tariffs. Computers & Industrial Engineering, 112, 721-734.
[7] Cota, L. P., Coelho, V. N., Guimarães, F. G., & Souza, M. J. (2018). Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times. International Transactions in Operational Research.
[8] Da Col, G., & Teppan, E. C. (2019). Industrial Size Job Shop Scheduling Tackled by Present Day CP Solvers. In International Conference on Principles and Practice of Constraint Programming (pp. 144-160). Springer, Cham.
[9] Ding, J. Y., Song, S., Zhang, R., Chiong, R., & Wu, C. (2015). 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.
[10] 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(1-2), 199-227.
[11] Graham, R. L. (1966). Bounds for certain multiprocessing anomalies. Bell System Technical Journal, 45(9), 1563-1581.
[12] Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1988). A general algorithm for scheduling batch operations. In PSE'88: Third International Symposium on Process Systems Engineering: In Affiliation with CHEMECA 88, a Bicentennial Event; Sydney, Australia, 28 August-2 September, 1998; Preprints of Papers (p. 62). Institution of Engineers, Australia.
[13] Kurniawan, B., Gozali, A. A., Weng, W., & Fujimura, S. (2017, December). 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). IEEE.
[14] Laborie, P., & Rogerie, J. (2016). Temporal linear relaxation in IBM ILOG CP Optimizer. Journal of Scheduling, 19(4), 391-400.
[15] Manne, A. S. (1960). On the job-shop scheduling problem. Operations Research, 8(2), 219-223.
[16] Meng, L., Zhang, C., Shao, X., & Ren, Y. (2019). MILP models for energy-aware flexible job shop scheduling problem. Journal of Cleaner Production, 210, 710-723.
[17] Mitra, S., Grossmann, I. E., Pinto, J. M., & Arora, N. (2012). Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes. Computers & Chemical Engineering, 38, 171-184.
[18] 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(1-4), 523-535.
[19] NY Times (2019) nytimes.com/2019/11/21/nyregion/nyc-building-grades-letters.html
[20] Rubaiee, S., Cinar, S., & Yildirim, M. B. (2018). An energy-aware multiobjective optimization framework to minimize total tardiness and energy cost on a single-machine nonpreemptive scheduling. IEEE Transactions on Engineering Management, 66(4), 699-714.
[21] Sharma, A., Zhao, F., & Sutherland, J. W. (2015). Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff. Journal of Cleaner Production, 108, 256-270.
[22] 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.
[23] Storer, R. H., Wu, S. D., & Vaccari, R. (1992). New search spaces for sequencing problems with application to job shop scheduling. Management science, 38(10), 1495-1509.
[24] Tan, M., Duan, B., & Su, Y. (2018). Economic batch sizing and scheduling on parallel machines under time-of-use electricity pricing. Operational Research, 18(1), 105-122.
[25] 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.
[26] Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. Naval Research Logistics Quarterly, 6(2), 131-140.
[27] Wang, S., Zhu, Z., Fang, K., Chu, F., & Chu, C. (2018). Scheduling on a two-machine permutation flow shop under time-of-use electricity tariffs. International Journal of Production Research, 56(9), 3173-3187.
[28] Wang, Y., & Li, L. (2013). Time-of-use based electricity demand response for sustainable manufacturing systems. Energy, 63, 233-244.
[29] Wu, P., Cheng, J., & Chu, F. (2019). Large-scale energy-conscious bi-objective single-machine batch scheduling under time-of-use electricity tariffs via effective iterative heuristics. Annals of Operations Research, 1-24.
[30] Zeng, Y., Che, A., & Wu, X. (2018). Bi-objective scheduling on uniform parallel machines considering electricity cost. Engineering Optimization, 50(1), 19-36.
[31] Zhang, H., Fang, Y., Pan, R., & Ge, C. (2018). A new greedy insertion heuristic algorithm with a multi-stage filtering mechanism for energy-efficient single machine scheduling problems. Algorithms, 11(2), 18.
[32] Zhang, S., Che, A., Wu, X., & Chu, C. (2018). 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.

BIOGRAPHIES

Andy Ham received Ph.D. in industrial engineering from Arizona State University in 2009, and M.S. in OR/IE from University of Texas at Austin in 2000.

He is currently working as a professor in Industrial Engineering, Liberty University, Virginia, while he serves as a technical advisor for semiconductor manufacturers and CSPI. Prior to the current position, he worked for BerkshireGrey, Samsung Electronics, Samsung Austin Semiconductor, Global Foundries, AMD, and IBM/ILOG. His research is currently focusing on real-time scheduling of vehicles, drones and robots in smart factories, smart warehouses, and logistics industry.