

Lingxuan LIU, Zhongshun SHI, Leyuan SHI

Minimization of total energy consumption in an m-machine flow shop with an exponential time-dependent learning effect

© The Author(s) 2018. Published by Higher Education Press. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>)

Abstract This study investigates an energy-aware flow shop scheduling problem with a time-dependent learning effect. The relationship between the traditional and the proposed scheduling problem is shown and objective is to determine a job sequence in which the total energy consumption is minimized. To provide an efficient solution framework, composite lower bounds are proposed to be used in a solution approach with the name of Bounds-based Nested Partition (BBNP). A worst-case analysis on shortest process time heuristic is conducted for theoretical measurement. Computational experiments are performed on randomly generated test instances to evaluate the proposed algorithms. Results show that BBNP has better performance than conventional heuristics and provides considerable computational advantage.

Keywords flow shop, energy-aware scheduling, learning effect, nested partition, worst-case error bound

1 Introduction

Conventional scheduling problems with time-based objectives have been widely investigated in the past few decades. Conventional scheduling problems that are usually embedded in a synthetic industrial information

Received April 28, 2018; accepted August 1, 2018

Lingxuan LIU
Department of Industrial Engineering & Management, Peking University, Beijing 100871, China

Zhongshun SHI
Department of Industrial & Systems Engineering, University of Wisconsin-Madison, WI, USA

Leyuan SHI (✉)
Department of Industrial Engineering & Management, Peking University, Beijing 100871, China; Department of Industrial & Systems Engineering, University of Wisconsin-Madison, WI, USA
E-mail: leyuan@coe.pku.edu.cn

system deal with the allocation of jobs on machines over a given time period. The objective is focused on the reduction of production time, which is related to turnover rate and appropriate economic indicators. However, in terms of sustainable development, the consumed energy in scheduling has become a public economic benefit for manufacturing companies with the continuous increase of environmental issues and large amount of energy costs. By contrast, the processing time of a job or an operation is constantly fixed through the entire production for the classical research on scheduling. However, there are many realistic manufacturing environments, the labor or production facility improves continuously over time and the learning effect continuously improve over time, which indicate that a job has a shorter processing time than that of previously scheduled jobs that is an extremely important factor to consider. This condition provides an accurate estimation on energy consumption, which causes better calculation of processing time of workers and machines. Thus, integrating the learning effect to the energy-based scheduling objective will aid in actual processing time approximation and accurate estimation of total energy consumption. For the close relationship between the energy-aware scheduling and scheduling with learning effects, an integrated research on the combined problem should be conducted regardless of its immense complexity.

Despite various studies on the traditional flow shop scheduling problem (FSS), the literature on scheduling problems that consider sustainable factors is limited, and the integrated scheduling problem with simultaneous considerations of energy reduction and learning effects has not been investigated due its high complexity. In this study, we first propose an energy-aware FSS (EAFSS) problem with a time-dependent learning effect. A new efficient framework with bound-based selection, partitioning, and sampling-based strategy is designed to solve the combined problem. This framework can be practically extended to solve other combinatorial problems where efficient bounds can be developed. A bound-based nested

partition (BBNP) with a composite bound is proposed to obtain an optimal solution. We also conduct a worst-case analysis on a shortest processing time (SPT) algorithm for theoretical measurement.

The remainder of this paper is organized as follows: Section 2 presents a literature review in terms of FSS with learning effect (FSSLE) and energy-aware scheduling (EAS). Section 3 provides the notations and assumptions. Section 4 shows the relationship between the traditional and proposed problems. Section 5 develops a composite bound for sustainable FSS with the exponential sum-of-processing-time-based learning effects. Sections 6 and 7 describe the proposed composite solution and worst-case analysis on SPT heuristic. Section 8 presents the detailed experimental results and analysis. Section 9 summarizes the conclusions.

2 Literature review

As a conventional topic, FSS has been intensively investigated in the past few decades. Johnson (1954), Gonzalez and Sahni (1978), Garey and Johnson (1979), and Smutnicki (1998) investigated typical flow shop models and their associated algorithms. Polynomial solutions can be found in several certain cases (Johnson, 1954). However, in most cases, it is not able to be solved in polynomial time (Gonzalez and Sahni, 1978), which leads to the study on approximation algorithms (Garey and Johnson, 1979; Smutnicki, 1998). Many real applications have supported Wright's theory (1936) where the processing time can be remarkably reduced with the increase of production requirements (Webb, 1994). Biskup (1999) and Cheng and Wang (2000) successfully introduced Wright's theory in scheduling. Considerable studies have focused on FSSLE. Lee and Wu (2004) investigated a two-machine FSS with position-based learning effects. They proposed heuristic and branch-and-bound methods to minimize the total completion time. Recently, Pei et al. (2017) evaluated a serial-batching scheduling problem with position-based learning effect under single and parallel machine environments, which minimizes the total number of delayed jobs and maximizes the earliness. They developed an optimization algorithm for a single machine setting and proposed a hybrid BA-VNS algorithm for a parallel machine setting. For a learning-based scheduling environment, Wang and Xia (2005) investigated FSS to minimize either the total flow time or makespan. They determined polynomial time solutions in several certain cases. They proposed a heuristic algorithm and confirmed the worst-case error bound of the heuristic. Koulamas and Kyparisis (2007) considered sum-of-job-processing-time-based and position-based learning effects in a two-machine FSS and showed that the SPT sequence is optimal under several certain cases. On the basis of the similar concept of Wang and Xia (2005), Xu

et al. (2008) implemented the worst-case analysis on SPT for the FSSLE. Their objectives are all related to completion time. Pei et al. (2017) assessed a serial-batch scheduling problem in an environment where coordination exists among multiple factories and resource-dependent processing time and dual constraints of resources are considered. They provided a polynomial-time scheduling rule for the optimization of each single factory and proposed an efficient BA-VNS algorithm to solve the complex coordinate problem. Wu and Lee (2009), Rudek (2011), Kuo et al. (2012), Sun et al. (2013), Wang et al. (2013), Xu et al. (2016), and Gao et al. (2017) conducted recent studies on FSSLE.

EAS is an area that has been rarely investigated in the scheduling domain regardless of its importance. Liu et al. (2001), Artigues et al. (2009), Lee and Zomaya (2010), Chan et al. (2013), and Agrawal and Rao (2014) conducted previous studies associated on EAS. Liu et al. (2001) considered power-aware scheduling under timing-constraints. Artigues et al. (2009) investigated a scheduling problem with energy constraints. Tree searches are applied to provide schedule solutions. Lee and Zomaya (2010) evaluated a scheduling framework that incorporates various factors related with energy-efficient operation. Chan et al. (2013) introduced a scheduling problem with weighted flow time and energy graphs. Algorithms with the basic concept of removing certain tasks and maintaining the optimality are developed. Agrawal and Rao (2014) conducted a systematic study on EAS of a distributed system. They proved that total energy consumption minimization is equivalent to makespan minimization for several special cases. In addition, several heuristics are developed for the solution. Liu et al. (2017) considered an EAS problem with processing-speed-dependent processing times and proposed a three-stage decomposition approach to minimize energy consumption.

3 Notations and assumptions

The investigated problem here includes n jobs $J = \{J_1, J_2, \dots, J_n\}$, where each job in the system should be sequentially processed on each of m machines $M = \{M_1, M_2, \dots, M_m\}$ and without preemption. The sequence of the job remains the same for each machine, which indicates that each job joins the queue of the next machine after its completion on its own machine and all the queues are assumed to follow a first in first out principle. Similar to the study of Agrawal and Rao (2014), the energy consumption in the working and idle states for each machine M_i are denoted as $\mu(M_i)$ and $k\mu(M_i)$, respectively, where $0 \leq k \leq 1$. Each job J_j consists m operations $O_{1j}, O_{2j}, \dots, O_{mj}$, and these operations should be sequentially processed on each machine. p_{ij} denotes the (original) processing time of operation O_{ij} .

A previous study on the learning effect assumed that the learning curve is position-based, which is a log-linear learning curve $p_{jr} = p_j r^a$ where a is a constant index and is position in a schedule. The processing time exponentially decreases with the increase of position r when $a < 0$. An exponential model of learning effect $p_{jr} = p_j \alpha^{r-1}$, where $0 \leq \alpha \leq 1$, is proposed by Wang and Xia (2005). The effect is constant for a given job because $p_{j,r+1} p_{j,r} = \alpha$. With this approach, the processing time decreases at a reasonably slow rate when α is close to 1. This condition is suitable for several manufacturing environments where the learning effect is slow.

Motivated by this condition, we consider an exponential time-dependent learning effect in this study, and actual processing time p_{ijr} is defined as follows:

$$p_{ijr} = p_{ij} (\alpha \sum_{l=1}^{r-1} p_{ij}^l + \beta), \quad i = 1, 2, \dots, m; \\ r, j = 1, 2, \dots, n, \quad (1)$$

where $0 < \alpha \leq 1$ is the learning rate, $\alpha \geq 0$, $\beta \geq 0$, and $\alpha + \beta = 1$.

For a given schedule π , the completion time of operation O_{ij} is denoted by $C_{ij}(\pi) = C_{ij}$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$, and $C_j(\pi) = C_j = C_{mj}$ represents the completion time of job J_j . $\tau_c(M_i)$ denotes the total working time on machine M_i , and $\tau_d(M_i)$ represents the total idle time on machine M_i . Thus, $\tau_d(M_i) = C_{\max} - \tau_c(M_i)$. This condition aims to determine a schedule to minimize the total energy consumption, which is expressed as $E = \sum_{i=1}^m [\mu(M_i) \tau_c(M_i) + k \mu(M_i) \tau_d(M_i)]$. The problem can be written as $Fm | p_{ijr} = p_{ij} (\alpha \sum_{l=1}^{r-1} p_{ij}^l + \beta) | \sum_{i=1}^m [\mu(M_i) \tau_c(M_i) + k \mu(M_i) \tau_d(M_i)] (m \geq 3)$ and it is well known that this problem is NP-complete.

4 Analysis on minimum energy consumption scheduling with learning effects

Considering that $\tau_d(M_i) = C_{\max} - \tau_c(M_i)$, total energy consumption E can also be expressed as

$$E = \sum_{i=1}^m [\mu(M_i) \tau_c(M_i) + k \mu(M_i) \tau_d(M_i)] \\ = (1-k) \sum_{i=1}^m \mu(M_i) \tau_c(M_i) + k \sum_{i=1}^m \mu(M_i) C_{\max},$$

where k and $\mu(M_i)$ are fixed constants. The traditional assumption on the objective of the scheduling problem focuses on the time reduction of tasks and usually includes the minimization of makespan, maximum delay, total completion time, and total tardiness. This assumption is

further related to overall turnover rate or customer satisfaction, which are important economic and management factors. The objective considered in this study provides such a sustainable way of scheduling where the total energy consumption is kept low.

On the basis of the above equation, when the learning effect is ignored (processing time is assumed to be constant) and power consumption rates $\mu(M_i)$ are equal for all the machines, that is, $\mu(M_1) = \mu(M_2) = \dots = \mu(M_m) = \mu(M)$, we can have

$$E = (1-k) \sum_{i=1}^m \mu(M_i) \tau_c(M_i) + k \sum_{i=1}^m \mu(M_i) C_{\max} \\ = (1-k) \mu(M) \sum_{i=1}^m \tau_c(M_i) + k \sum_{i=1}^m \mu(M_i) C_{\max}.$$

Considering that k , $\mu(M_i)$, and $\sum_{i=1}^m \tau_c(M_i)$ are all fixed constants, the problem will be reduced to the minimization of makespan. For the general case, the objective is the addition of two factors, namely, minimum working energy consumption and minimum makespan. The setting of different energy consumption rates for different machines lead to the tradeoffs between economic and sustainable factors because makespan is used as an important corporate economic indicator. In addition, the entire consumed time can be controlled within a reasonable limit by reducing the energy consumption although the minimum energy consumption cannot guarantee the minimum makespan.

Hence, the sustainable perspective does not conflict with the corporate economic profits or management level in terms of EAS for the operation management of manufacturers. On the contrary, a good control of consumed energy can have a remarkable impact on the reduction of production time and costs. The benefit of energy-aware objective is that it is a generalization of minimizing production time, which can lead to a better balance. The different settings of energy consumption rates provide opportunities in configuring the problem based on various preferences.

5 Composite lower bounds (LBs)

In this section, we propose a composite bound on partial solutions under three different perspectives, namely, $LB(1)$, $LB(2)$, and $LB(3)$, which is applied to reduce the total nodes searched in a branch-and-bound framework. Let $\pi = (PS, US)$ be a schedule of jobs where PS and US represent the scheduled and unscheduled parts, respectively.

Lemma 1. (Wang et al. (2009)) For problem 1 $| p_{jr} = p_j (\alpha \sum_{l=1}^{r-1} p_{ij}^l + \beta) | C_{\max}$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of p_j (i.e., the SPT rule).

5.1 LB 1

First, we consider an LB for the term C_{\max} . For a partial schedule with k scheduled jobs, the completion time of the $(k + 1)$ th job on machine M_i can be written as

$$C_{i[k+1]}(\pi) = \max\{C_{i-1,[k+1]}(\pi), C_{i[k]}(\pi)\} + p_{i[k+1]}(\alpha a \sum_{l=1}^k P_{i[l]} + \beta) \geq C_{i[k]}(\pi) + p_{i[k+1]}(\alpha a \sum_{l=1}^k P_{i[l]} + \beta).$$

Similarly,

$$C_{m[k+1]}(\pi) \geq C_{i[k]}(\pi) + \sum_{u=i}^m P_{u[k+1]}(\alpha a \sum_{l=1}^k P_{u[l]} + \beta),$$

$$C_{m[k+j]}(\pi) \geq C_{i[k]}(\pi) + \sum_{u=i}^{m-1} P_{u[k+1]}(\alpha a \sum_{l=1}^k P_{u[l]} + \beta) + \sum_{h=1}^j P_{m[k+h]}(\alpha a \sum_{l=1}^k P_{m[l]} + \sum_{l=1}^{h-1} P_{m[k+l]} + \beta),$$

where $1 \leq j \leq n - k$.

Hence, the makespan is

$$C_{\max} = C_{[n]}(\pi) \geq C_{i[k]}(\pi) + \sum_{u=i}^{m-1} P_{u[k+1]}(\alpha a \sum_{l=1}^k P_{u[l]} + \beta) + \sum_{h=1}^{n-k} P_{m[k+h]}(\alpha a \sum_{l=1}^k P_{m[l]} + \sum_{l=1}^{h-1} P_{m[k+l]} + \beta). \quad (2)$$

The first term is a constant, and an LB can be obtained by minimizing the last two terms. From Lemma 1, the term $\sum_{h=1}^{n-k} P_{m[k+h]}(\alpha a \sum_{l=1}^k P_{m[l]} + \sum_{l=1}^{h-1} P_{m[k+l]} + \beta)$ can be minimized by sequencing the jobs in non-decreasing order of $P_{m[j]}$. Thus, the LB for C_{\max} concerning machine M_i is

$$LB_i(C_{\max}) = C_{i[k]}(\pi) + \sum_{u=i}^{m-1} P_{u[k+1]}^*(\alpha a \sum_{l=1}^k P_{u[l]} + \beta) + \sum_{h=1}^{n-k} P_{m[k+h]}(\alpha a \sum_{l=1}^k P_{m[l]} + \sum_{l=1}^{h-1} P_{m[k+l]} + \beta), \quad (3)$$

where $[k + 1]^* = \arg \min_{[k+1]} \sum_{u=i}^{m-1} P_{u[k+1]}$.

For the term $\sum_{i=1}^m \mu(M_i)\tau_c(M_i)$, we can easily show that

$$\sum_{i=1}^m \mu(M_i)\tau_c(M_i) \geq \mu_{\min} \sum_{i=1}^m \sum_{j=1}^k P_{i[j]}(\alpha a \sum_{l=1}^{j-1} P_{i[l]} + \beta) + \mu_{\min} \sum_{h=1}^{n-k} L_{(k+h)}(\alpha a \sum_{i=1}^m \sum_{l=1}^k P_{i[l]} + \sum_{l=1}^{h-1} L_{(k+l)} + \beta),$$

where μ_{\min} is the minimum value among $\mu_i (1 \leq i \leq m)$, and $L_{(k+1)} \leq L_{(k+2)} \leq \dots \leq L_{(n)}$ is a nondecreasing order of total normal processing times of unscheduled jobs.

In addition,

$$\sum_{i=1}^m \mu(M_i)\tau_c(M_i) \geq \mu_{\min} \sum_{i=1}^m \sum_{j=1}^k P_{i[j]}(\alpha a \sum_{l=1}^{j-1} P_{i[l]} + \beta) + \sum_{i=1}^m \sum_{h=1}^{n-k} P_{i(k+h)}(\alpha a \sum_{l=1}^k P_{i[l]} + \sum_{l=1}^{h-1} P_{i(k+l)} + \beta).$$

Thus, the LB for the term $\sum_{i=1}^m \mu(M_i)\tau_c(M_i)$ is

$$LB^1 = \min \left(\mu_{\min} \sum_{i=1}^m \sum_{j=1}^k P_{i[j]}(\alpha a \sum_{l=1}^{j-1} P_{i[l]} + \beta) + \mu_{\min} \sum_{h=1}^{n-k} L_{(k+h)}(\alpha a \sum_{i=1}^m \sum_{l=1}^k P_{i[l]} + \sum_{l=1}^{h-1} L_{(k+l)} + \beta), \mu_{\min} \sum_{i=1}^m \sum_{j=1}^k P_{i[j]}(\alpha a \sum_{l=1}^{j-1} P_{i[l]} + \beta) + \sum_{i=1}^m \sum_{h=1}^{n-k} P_{i(k+h)}(\alpha a \sum_{l=1}^k P_{i[l]} + \sum_{l=1}^{h-1} P_{i(k+l)} + \beta) \right).$$

Accordingly, $LB(1)_i = (1 - k)LB^1 + k \sum_{i=1}^m \mu(M_i)LB(1)_i(C_{\max})$. We select the maximum value of $LB(2)_i$ to obtain a tight LB, and $LB(1)$ is denoted as follows:

$$LB(1) = \max\{LB(1)_1, LB(1)_2, \dots, LB(1)_m\}. \quad (4)$$

5.2 LB 2

Following the similar perspective that each certain machine generates an LB is taken, we consider all the remaining operations of a job in the calculation of $C_{i[k+1]}(\pi)$ compared with LB 1. Thus, we have

$$C_{i[k+1]}(\pi) \geq C_{i[k]}(\pi) + p_{i[k+1]}(\alpha a \sum_{l=1}^k P_{i[l]} + \beta).$$

Similarly,

$$C_{m[k+1]}(\pi) \geq C_{i[k]}(\pi) + \sum_{u=i}^m p_{u[k+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \beta), \quad \geq C_{1[k]}(\pi) + \sum_{u=1}^i p_{u[k+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \beta).$$

$$C_{m[k+j]}(\pi) \geq C_{i[k]}(\pi) + \sum_{h=1}^j p_{i[k+h]}(\alpha \sum_{l=1}^k p_{i[l]} + \sum_{l=1}^{h-1} p_{i[k+l]} + \beta) \\ + \sum_{u=i+1}^m p_{u[k+j]}(\alpha \sum_{l=1}^k p_{u[l]} + \sum_{l=1}^{j-1} p_{u[k+l]} + \beta),$$

where $1 \leq j \leq n-k$.

Hence, the makespan is

$$C_{max} = C_{[n]}(\pi) \geq C_{i[k]}(\pi) \\ + \sum_{h=1}^{n-k} p_{i[k+h]}(\alpha \sum_{l=1}^k p_{i[l]} + \sum_{l=1}^{h-1} p_{i[k+l]} + \beta) \\ + \sum_{u=i+1}^m p_{u[n]}(\alpha \sum_{l=1}^{n-1} p_{u[l]} + \beta). \quad (5)$$

The first term is a constant, and the minimization of the last two terms will lead to a low bound. From Lemma 1, the term $\sum_{h=1}^{n-k} p_{i[k+h]}(\alpha \sum_{l=1}^k p_{i[l]} + \sum_{l=1}^{h-1} p_{i[k+l]} + \beta)$ can be minimized by sequencing the jobs in non-decreasing order of $p_{i[j]}$. Thus, the LB for machine M_i is

$$LB_i(C_{max}) = C_{i[k]}(\pi) \\ + \sum_{h=1}^{n-k} p_{i[k+h]}(\alpha \sum_{l=1}^k p_{i[l]} + \sum_{l=1}^{h-1} p_{i[k+l]} + \beta) \\ + \sum_{u=i+1}^m p_{u[n]^*}(\alpha \sum_{l=1}^{n-1} p_{u[l]} + \beta), \quad (6)$$

where $[n]^* = \arg \min_{[n]} \sum_{u=i+1}^m p_{u[n]^*}$.

Accordingly, $LB(2)_i = (1-k)LB^1 + k \sum_{i=1}^m \mu(M_i)LB(2)_i(C_{max})$. We select the maximum value of $LB(2)_i$ to obtain a tight LB, and $LB(2)$ is denoted as follows:

$$LB(2) = \max\{LB(2)_1, LB(2)_2, \dots, LB(2)_m\}. \quad (7)$$

5.3 LB 3

We take a new perspective for each job in the calculation of LB 3. $C_{i[k+1]}(\pi)$ is calculated based on the arbitrary position of a certain job.

$$C_{i[k+1]}(\pi) = \max\{C_{i-1, [k+1]}(\pi), C_{i[k]}(\pi)\} \\ + p_{i[k+1]}(\alpha \sum_{l=1}^k p_{i[l]} + \beta)$$

Similarly,

$$C_{m[k+1]}(\pi) \geq C_{1[k]}(\pi) + \sum_{u=1}^m p_{u[k+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \beta),$$

$$C_{i[k+j]}(\pi) \geq C_{1[k]}(\pi) + \sum_{h=1}^{j-(w+1)}$$

$$p_{i[k+w+h+1]}(\alpha \sum_{l=1}^k p_{i[l]} + \sum_{l=1}^{w+1} p_{i[k+l]} + \sum_{l=1}^{h-1} p_{i[k+w+l+1]} + \beta)$$

$$+ \sum_{h=1}^w p_{1[k+h]}(\alpha \sum_{l=1}^k p_{1[l]} + \sum_{l=1}^{h-1} p_{1[k+l]} + \beta)$$

$$+ \sum_{u=1}^i p_{u[k+w+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \sum_{l=1}^w p_{u[k+l]} + \beta)$$

where $1 \leq j \leq n-k$, $1 \leq w \leq j-k$.

$$C_{m[k+j]}(\pi) \geq C_{1[k]}(\pi) + \sum_{h=1}^{j-(w+1)}$$

$$p_{m[k+w+h+1]}(\alpha \sum_{l=1}^k p_{m[l]} + \sum_{l=1}^{w+1} p_{m[k+l]} + \sum_{l=1}^{h-1} p_{m[k+w+l+1]} + \beta)$$

$$+ \sum_{h=1}^w p_{1[k+h]}(\alpha \sum_{l=1}^k p_{1[l]} + \sum_{l=1}^{h-1} p_{1[k+l]} + \beta)$$

$$+ \sum_{u=1}^m p_{u[k+w+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \sum_{l=1}^w p_{u[k+l]} + \beta),$$

where $1 \leq j \leq n-k$, $1 \leq w \leq j-k$.

Hence, the makespan is

$$C_{max} = C_{[n]}(\pi) \geq C_{1[k]}(\pi) + \sum_{h=1}^{n-(k+w+1)}$$

$$p_{m[k+w+h+1]}(\alpha \sum_{l=1}^k p_{m[l]} + \sum_{l=1}^{w+1} p_{m[k+l]} + \sum_{l=1}^{h-1} p_{m[k+w+l+1]} + \beta)$$

$$+ \sum_{h=1}^w p_{1[k+h]}(\alpha \sum_{l=1}^k p_{1[l]} + \sum_{l=1}^{h-1} p_{1[k+l]} + \beta)$$

$$+ \sum_{u=1}^m p_{u[k+w+1]}(\alpha \sum_{l=1}^k p_{u[l]} + \sum_{l=1}^w p_{u[k+l]} + \beta), \quad (8)$$

where $1 \leq w \leq n-k$. The first term is a constant, and the minimization of the last three terms will lead to an LB.

Moreover, the second term is minimized by sequencing all jobs in a non-decreasing order of $p_{\min(j)} = \min\{p_{1[j]}, p_{m[j]}\}$ based on Lemma 1, except for the $(k + w + 1)$ th job. By contrast, the actual processing time can be minimized by increasing the accumulated learning effect. Hence, we take $p_{\max(j)} = \min\{p_{1[j]}, p_{m[j]}\}$ as the processing time of the past position. Consequently, the LB for job J_j is

$$LB_j(C_{\max}) = C_{1[k]}(\pi) + \sum_{u=1}^m p_{uj}(\alpha \sum_{l=1}^k p_{u[l]} + \sum_{l=1}^{n-k-1} p_{u[k+l]} + \beta) + \sum_{h=1}^{n-k-1} p_{\min(k+h)}(\alpha \sum_{l=1}^k p_{1[l]} + \sum_{l=1}^{h-1} p_{\max(k+h)} + \beta), \quad (9)$$

where $k \leq j \leq n - k$, $k \leq h \leq n - k$.

Accordingly, $LB(3)_j = (1 - k)LB^1 + k \sum_{i=1}^m \mu(M_i)LB(3)_i(C_{\max})$. We use the maximum value of $LB(2)_i$ to obtain a tight LB, and $LB(1)$ is denoted as follows:

$$LB(3) = \max\{LB(3)_1, LB(3)_2, \dots, LB(3)_{n-k}\}. \quad (10)$$

Three LBs are derived from different perspectives, and these LBs form the composite bound by using their maximum value. Thus, the composite bound is denoted as

$$LB = \max\{LB(1), LB(2), LB(3)\}. \quad (11)$$

6 Algorithms

6.1 Two heuristic algorithms

Considering that the proposed problem on m -machine ($m \geq 3$) flow shop is NP-complete, heuristic algorithms can be constructed to solve it within a limited amount of time. In this study, we use two heuristics, namely, $O(mn^2)$ NEH heuristic of Nawaz et al. (1983) and $O(mn^4)$ FL heuristic of Framinan and Leisten (2003) to solve $Fm|p_{ijr} = p_{ij}(\alpha \sum_{l=1}^{r-1} p_{i[l]} + \beta) | \sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$ ($m \geq 3$). They are widely applied and shown to be efficient in solving FSS. Now, we provide the modified NEH and FL algorithms with time complexity of $O(mn^3)$ and $O(mn^4)$, respectively.

NEH algorithm

Step 1. Calculate total normal processing time L_j of each job J_j .

Step 2. Denote the SPT sequence π , where all jobs are sorted in non-decreasing order of L_j , and let π_k be job in the k th position of π .

Step 3. Select π_1, π_2 from π , switch their positions, calculate the energy consumption of the two possible sequences, and select the better one. The relative positions of the two jobs will not change in the remaining steps. Set $j = 3$.

Step 4. Use π_j , insert it in j possible positions of the partial sequence in the previous steps and result in j sequences, and calculate the energy of these sequences to find the best sequence. Keep the relative positions of jobs in the current best sequence unchanged.

Step 5. If $j = n$, STOP; otherwise set $j = j + 1$ and go to Step 4.

FL algorithm

Step 1. Calculate total normal processing time L_j of each job J_j .

Step 2. Denote the SPT sequence π , where all jobs are sorted in non-decreasing order of L_j , and denote π_k the job in the k th position of π .

Step 3. Set $k = 2$. Pick π_1, π_2 from π , switch their positions, calculate the energy consumption of the two possible sequences, and select the better one. The relative positions of the two jobs will not change in the remaining steps.

Step 4. Set $k = k + 1$. Pick π_j , insert it in k possible positions of the partial sequence found in the previous steps and result in k sequences, calculate the energy of these sequences to find the best sequence, and go to Step 5.

Step 5. Interchange jobs in all possible positions i and j of the above best sequence, where $1 \leq i < k, i < j \leq k$. This process will result in $k(k - 1)/2$ sequences, and select the one with minimum energy consumption as the best partial sequence.

Step 6. If $k = n$, STOP; otherwise, go to Step 4.

6.2 BBNP algorithm

On the basis of the abovementioned studies, we propose the BBNP algorithm in this subsection. We combine the above composite LB with Nested Partition algorithm. Shi and Ólafsson (2000) proposed the NP algorithm, where they optimized the allocation of computational effort during searching and sustained the global view, in which considerable computational efforts are allocated to the most promising region. The NP algorithm has been successfully applied in a wide range of combinatorial and large-scale optimization problems (Shi et al. 2001; Wu et al. 2010)

For each iteration of a general NP framework, the most promising region is divided into small subregions (partitioning) and the remaining parts are aggravated as surrounding regions. Then, samples are randomly selected from subregions and surrounding regions (random sampling). Subsequently, the promising index is calculated for each region, and this promising index will serve as the guidance direction for the next iteration. Finally, the movement is performed based on the promising index.

However, good implementation of the partition and sampling procedure is difficult for the NP method. For the mitigation and improvement of the performance of the algorithm, we introduce the LB and upper bound (UB) in

the solution, which can provide a mathematical-based strategy of partitioning and sampling and a good guide for the implementation.

Bounds-based-partitioning (BBP)

Partitioning is a key step used in BBNP. For partitioning, the promising region is divided into several small subregions. Given a huge feasible solution space, the underlying problem is that the computational effort should not be wasted in the subregions that are not able to contribute to optimum searching. By contrast, BBNP trims the subregion that cannot generate the optimal solution and reduces the searching region to a small scale. To achieve this condition, a good UB should be determined, calculate the LB for $\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$ of the unfathomed partial schedules, and trim the useless regions with the UB smaller than LB. The sampling procedure is only implemented in the remaining subregion. Figure 1 illustrates the BBP procedure.

BBP procedure

Step 1. Obtain the UB of BBNP by implementing the heuristic algorithms and select the best solution generated.

Step 2. In the k th level, divide promising region θ into $n-k$ subregions by partitioning.

Step 3. Calculate the LB for unfathomed partial schedules or C_{max} of the completed schedules for each obtained subregion.

Step 4. Trim the useless subregions with UB smaller than LB and obtain $M_{\sigma(k)}$ subregions. Denote $\sigma_1(k), \sigma_2(k), \dots, \sigma_{M_{\sigma(k)}}(k)$ as the final subregions to be investigated.

Bound-based-sampling (BBS)

Sampling should be implemented for each subregion acquired in the above procedure. An efficient sampling strategy provides an actual reflection of the subregion and has a direct impact on the performance of BBNP. The solution obtained from the sampling can be used as the evaluation factor and the new UB when it is smaller than the current UB. Similar to the previous procedure, we introduce bounds to trim the nodes that are unlikely to lead to optimum. The BBS procedure is shown in Fig. 2.

BBS procedure

Step 1. For subregion $\sigma_j(k)$ to be sampled, the $(k+1)$ th

level node is fixed by BBP procedure. Calculate the LB for each of the rest $n-k-1$ levels.

Step 2. In the $(k+2)$ th level, trim its subregion with UB smaller than LB, and implement a random selection in the remaining subregions to allocate job on the $(k+2)$ th level node.

Step 3. repeat Step 2 in $(k+3)$ th, $(k+4)$ th, ..., $(n-1)$ th level.

To avoid duplicate sampling, construct a new archive to store history sample nodes in each BBS procedure, and this archive is used to compare with new samples.

Estimating the promising index and backtracking

This procedure is the same with conventional NP framework and the promising index is calculated by the best objective value determined in a particular region.

The full BBNP algorithm is described as follows:

BBNP algorithm

Step 1. Denote d as the algorithm level, t as the run time, and set $d = 1$ and $t = 0$.

Step 2. If $d > n$ or $t > \text{TIMELIMIT}$, then STOP.

Step 3. Implement BBP procedure to divide root region θ into n initial subregions without generating any surrounding region.

Step 4. Use the BBS procedure in every current region.

Step 5. Calculate the promising index. If the best promising index exists in a subregion σ^* , go to Step 6; otherwise, go to Step 7.

Step 6. Select σ^* as the next promising region. Generate the surrounding region with other solution space. Set $d = d + 1$ and $t = \text{CURRENTTIME}$.

Step 7. Backtrack to the parent region of the current promising region. Set $d = d - 1$ and $t = \text{CURRENTTIME}$.

7 Worst-case bounds

On the basis of Lemma 1, we can build an approximate solution by using the SPT rule to solve $Fm|p_{ijr} = p_{ij}(a\alpha\sum_{i=1}^{r-1} p_{i|l} + \beta)|\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$.

Theorem 1. Let π^* be an optimal schedule and π be an

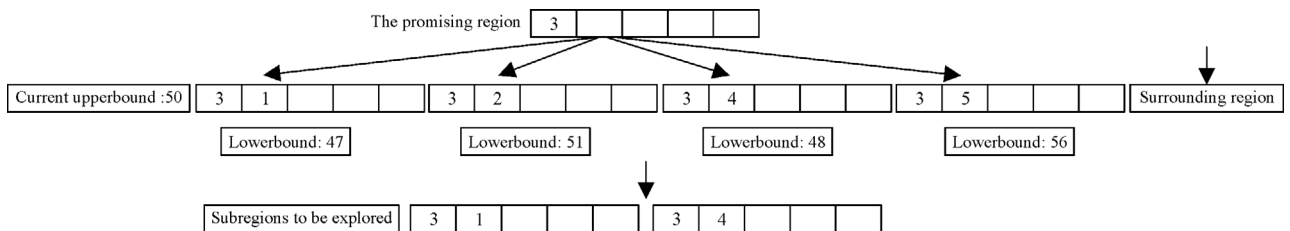


Fig. 1 BBP procedure

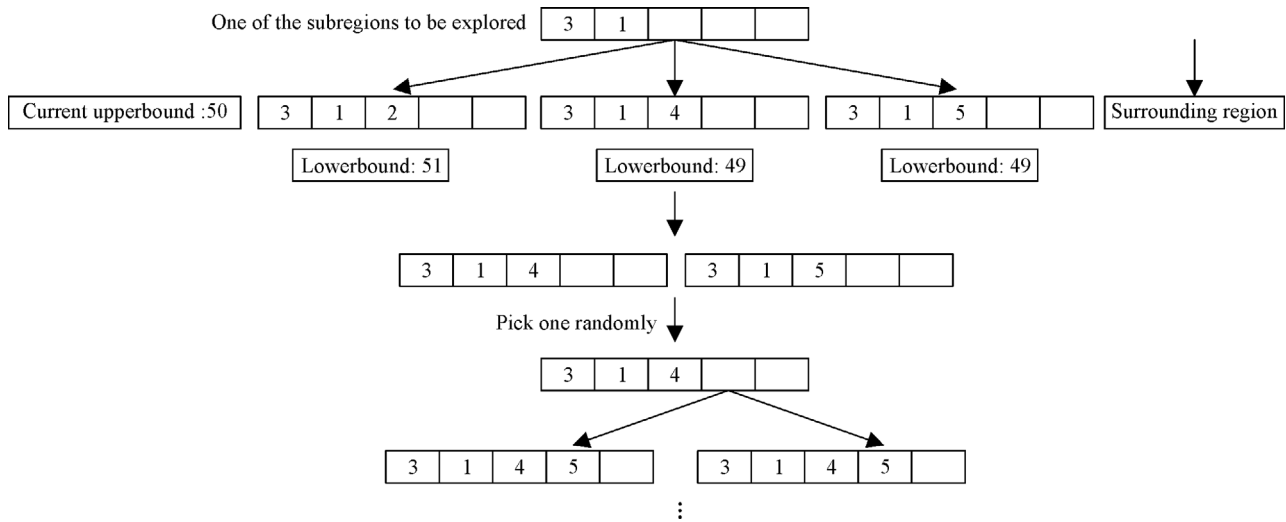


Fig. 2 BBS procedure

SPT schedule for $Fm|p_{ijr} = p_{ij}(\alpha a^{\sum_{l=1}^{r-1} p_{il}} + \beta)| \sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$ problem. Then,

$$\frac{\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)](\pi)}{\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)](\pi^*)} \leq \frac{m}{(\alpha a^{P_{\max}} + \beta)} \frac{\mu_{\max}(1-k) + k \sum_{i=1}^m \mu(M_i)}{\mu_{\min}(1-k) + k \sum_{i=1}^m \mu(M_i)}, \quad (12)$$

where $P_{\max} = \max\{\sum_{l=1}^n p_{il} - p_{i\min} | i = 1, 2, \dots, m\}$, $P_{i\min} = \min\{p_{ij} | j = 1, 2, \dots, n\}$, $\mu_{\max} = \max_{i=1}^m \mu(M_i)$, and $\mu_{\min} = \min_{i=1}^m \mu(M_i)$.

Proof. Similar to the proof in Wang J B and Wang J J (2014) for an SPT schedule π , where $L_1 \leq L_2 \leq \dots \leq L_n$, we have

$$C_j(\pi) \leq L_1 + L_2(\alpha a^{P_{\min,1}} + \beta) + L_3(\alpha a^{P_{\min,1} + P_{\min,2}} + \beta) + \dots + L_j(\alpha a^{P_{\min,1} + P_{\min,2} + \dots + P_{\min,j-1}} + \beta) \leq \sum_{l=1}^j L_l, \quad (13)$$

where $P_{\min,j} = \min\{p_{ij} | i = 1, 2, \dots, m\}$. Thus,

$$C_{\max}(S) \leq \sum_{j=1}^n L_j. \quad (14)$$

Denote $\pi^* = (J_{[1]}, J_{[2]}, \dots, J_{[n]})$ as the optimal schedule, where $[j]$ is job that is sequenced at the j th position of π^* , we have

$$C_{1[j]} = p_{1[1]} + p_{1[2]}(\alpha a^{p_{1[1]} + \beta} + \dots + p_{1[j]}(\alpha a^{p_{1[1]} + p_{1[2]} + \dots + p_{1[j-1]} + \beta}),$$

$$C_{2[j]} \geq p_{2[1]} + p_{2[2]}(\alpha a^{p_{2[1]} + \beta} + \dots + p_{2[j]}(\alpha a^{p_{2[1]} + p_{2[2]} + \dots + p_{2[j-1]} + \beta)),$$

...

$$C_{m[j]} \geq p_{m[1]} + p_{m[2]}(\alpha a^{p_{m[1]} + \beta} + \dots + p_{m[j]}(\alpha a^{p_{m[1]} + p_{m[2]} + \dots + p_{m[j-1]} + \beta)).$$

Hence,

$$C_{[j]}(S^*) \leq \frac{1}{m} \sum_{l=1}^j L_{[l]}(\alpha a^{P_{\max}} + \beta), \quad (15)$$

where

$$P_{\max} = \max\{\sum_{l=1}^n p_{il} - p_{i\min} | i = 1, 2, \dots, m\}$$

and $P_{i\min} = \min\{p_{ij} | i = 1, 2, \dots, n\}$. Thus,

$$C_{\max}(S^*) \leq \frac{1}{m} \sum_{l=1}^n L_l(\alpha a^{P_{\max}} + \beta) = \frac{1}{m} \sum_{l=1}^n L_l(\alpha a^{P_{\max}} + \beta). \quad (16)$$

Consequently, from Eqs. (14) and (16), we have

$$\frac{\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)](\pi)}{\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)](\pi^*)}$$

$$\leq \frac{\mu_{\max}(1-k)\sum_{j=1}^n L_j + k\sum_{i=1}^m \mu(M_i)\sum_{j=1}^n L_j}{\mu_{\max}(1-k)\sum_{j=1}^n L_j \frac{m}{(\alpha\alpha^{P_{\max}} + \beta)} + k\sum_{i=1}^m \mu(M_i)\sum_{j=1}^n L_j \frac{m}{(\alpha\alpha^{P_{\max}} + \beta)}}$$

$$\leq \frac{m}{(\alpha\alpha^{P_{\max}} + \beta)} \frac{\mu_{\max}(1-k) + k\sum_{i=1}^m \mu(M_i)}{\mu_{\min}(1-k) + k\sum_{i=1}^m \mu(M_i)}.$$

8 Computational experiments

For the evaluation of the heuristics and BBNP, computational experiments are conducted in this section. The algorithms are coded in C++ and run on a Pentium 4 with 2 GB RAM personal computer. The test problems are generated based on Li et al. (2013) and Wang and Wang (2014), and their specifications are shown as follows:

- learning rate: $a = \{0.7, 0.8, 0.9\}$;
 - processing time: $p_{ij} \in U[1, 100]$;
 - energy consumption in the working state: $\mu(M_i) \in (1, 2)$;
 - factors: $\alpha = \beta = 0.5, k \in (0, 1)$.
- Medium size instances:
- machine number: $m = 3, 5$;

- job number: $n = 9, 10, 11, 12, 13, 14$;
- time limit of BBNP approach: 900 s.

Large size instances:

- machine number: $m = 5$;
- job number: $n = 20, 50, 100, 150, 200$;
- time limit of BBNP approach: 1800 s.

For each combination of n - a - m , 20 random instances are generated.

Tables 1 and 2 show the computation results of medium-size problems. The average and maximum error gaps are used to reflect the performance of heuristic algorithms. The error gap of the solution is calculated as follows:

$$(V_{\text{Heur}} - V_{\text{opt}}) / V_{\text{opt}} \times 100\%,$$

where V_{Heur} is the value $\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$ of the solution generated by the heuristic algorithms,

Table 1 Error gap (%) of BBNP and heuristic algorithms for $m = 3$

n	a	SPT		NEH		FL		BBNP	
		mean	max	mean	max	mean	max	mean	max
9	0.7	13.8	40	2.25	9.15	0.1	1.73	0	0
9	0.8	13.2	30.3	3.13	17.6	0.076	0.626	0	0
9	0.9	13.6	34.6	2.81	13.2	0.058	0.777	0	0.01
10	0.7	15.8	34.9	2.19	10	0.16	2.74	0	0.01
10	0.8	16.9	38.8	2.49	16.9	0.099	1.62	0	0.015
10	0.9	16.6	37	1.63	8.35	0.141	1.57	0	0
11	0.7	15.8	34.4	3.82	11.4	0.104	1.15	0.067	1.246
11	0.8	13.7	30.2	2.38	9.24	0.281	2.36	0	0.015
11	0.9	14.8	31.9	1.68	6.43	0.403	3.67	0.133	1.617
12	0.7	15.6	34.2	1.84	6.33	0.153	2.23	0.011	0.124
12	0.8	16.9	39.2	1.81	5.41	0.347	3.12	0.084	1.016
12	0.9	15.6	31.9	2.21	8.28	0.356	3.87	0.104	1.246
13	0.7	17.7	35.4	2.38	13.7	0.385	4.15	0.184	2.039
13	0.8	16.6	31.5	2.41	7.08	0.359	2.18	0.101	0.998
13	0.9	16.8	29.3	2.69	8.52	0.254	2.97	0.125	1.028
14	0.7	11.9	24.5	1.84	7.47	0.105	1.23	0.092	0.926
14	0.8	13.4	21.7	2.28	8.6	0.296	2.27	0.104	1.638
14	0.9	13.3	21.5	2.29	8.12	0.187	2.16	0.067	0.102

Table 2 Error gap (%) of BBNP and heuristic algorithms for $m = 5$

n	a	SPT		NEH		FL		BBNP	
		mean	max	mean	max	mean	max	mean	max
9	0.7	14.8	40.8	2.5	10.5	0.49	3.32	0.308	3.31
9	0.8	13.8	27.5	4.26	12.1	0.762	4.62	0.314	2.05
9	0.9	15.3	36.2	3.45	12.1	0.773	2.94	0.284	2.12
10	0.7	17.4	29.5	3.5	16.4	0.75	3.07	0.454	3.31
10	0.8	17.6	34.4	3.19	11.1	0.976	5.28	0.402	3.2
10	0.9	16.6	28.3	3.18	11.5	0.447	3.44	0.333	2.98
11	0.7	18	38.3	4.11	19	0.733	3.06	0.438	2.39
11	0.8	16.8	31.7	2.48	9.07	0.964	4.51	0.698	4.9
11	0.9	17.9	42.2	2.77	8.06	0.648	3.31	0.276	1.39
12	0.7	18.9	29	3.28	10.9	1.26	5.48	1.1	5.31
12	0.8	20.1	38.8	5.39	12.3	1.03	6.01	0.585	2.79
12	0.9	15.8	32.4	3.57	11.5	0.558	2.35	0.322	1.86
13	0.7	19.7	42.9	3.46	14.1	1.15	3.7	0.542	2.63
13	0.8	18.9	38.8	2.83	8.68	0.172	1.78	0.132	1.51
13	0.9	16.5	34.6	2.15	9.69	0.576	4.15	0.299	2.07
14	0.7	18.9	30.6	2.9	9	0.866	4.62	0.774	3.7
14	0.8	19.6	38.2	4.21	9.38	1.33	8.43	0.99	4.34
14	0.9	16.7	30.9	2.9	12.8	0.559	2.68	0.392	2.4

and V_{opt} is the value $\sum_{i=1}^m [\mu(M_i)\tau_c(M_i) + k\mu(M_i)\tau_d(M_i)]$ of the optimal schedule. From the results, FL is more efficient than SPT and NEH. The performance is enhanced when the new framework BBNP is applied.

The frequency of maximum value, which indicates the dominating frequency of one LB for the partial schedule, is recorded in Table 3 when the composite bound method is applied. The maximum value is defined as MN_i/TN where MN_i is the number in which LB i dominates the others and TN is total number of iterations for one run. For the composite bounds, the frequency of domination of each LB on the evaluation of the partial schedule is almost equal through the results of frequency of maximum value. Hence, no superiority is observed among the domination of each LB in the composite strategy.

Table 4 shows the computation results of large-sized problems. In this experiment, the average and maximum ratios $\frac{E(Heur)}{E(ARB)}$ are used as the indicator on the performance of heuristic algorithms, where E is the total energy, $Heur \in \{SPT, NEH, FL, BBNP\}$ and ARB is any busy schedule (Smutnicki 1998). The ARB algorithm is selected as the reference value because it is a simple and arbitrary algorithm. From the results, the BBNP algorithm is more efficient than FL and FL is the most efficient among all the heuristics for large-scale instances.

Table 3 Results of domination of different LB calculations for partial schedule ($m = 3$)

n	a	Frequency of maximum		
		LB (1)	LB (2)	LB (3)
9	0.7	0.36	0.37	0.27
9	0.8	0.39	0.33	0.28
9	0.9	0.33	0.34	0.33
10	0.7	0.34	0.31	0.35
10	0.8	0.29	0.35	0.36
10	0.9	0.35	0.31	0.34
11	0.7	0.25	0.41	0.34
11	0.8	0.42	0.35	0.23
11	0.9	0.31	0.33	0.36
12	0.7	0.27	0.35	0.38
12	0.8	0.33	0.25	0.42
12	0.9	0.4	0.36	0.24
13	0.7	0.35	0.27	0.38
13	0.8	0.32	0.36	0.32
13	0.9	0.35	0.35	0.3
14	0.7	0.41	0.3	0.29
14	0.8	0.31	0.35	0.34
14	0.9	0.21	0.42	0.37

Table 4 Computational results of the heuristics for large-size instances ($m = 5$)

n	a	$\frac{E(SPT)}{E(ARB)}$		$\frac{E(NEH)}{E(ARB)}$		$\frac{E(FL)}{E(ARB)}$		$\frac{E(BBNP)}{E(ARB)}$	
		mean	max	mean	max	mean	max	mean	max
		20	0.7	0.8211	0.9519	0.782	0.9242	0.724	0.8728
20	0.8	0.804	0.9259	0.7657	0.8989	0.7018	0.8396	0.688	0.8016
20	0.9	0.8107	0.9358	0.7721	0.9086	0.7028	0.8456	0.6916	0.8334
40	0.7	0.8214	0.9406	0.7823	0.9132	0.7027	0.8432	0.677	0.7954
40	0.8	0.7869	0.9366	0.7494	0.9093	0.726	0.8655	0.6751	0.8153
40	0.9	0.7973	0.9239	0.7594	0.897	0.6903	0.8432	0.6815	0.8358
60	0.7	0.8144	0.9409	0.7756	0.9135	0.7027	0.8524	0.6885	0.8251
60	0.8	0.8096	0.9456	0.771	0.9181	0.7022	0.8531	0.6751	0.8165
60	0.9	0.8073	0.9125	0.7689	0.886	0.6901	0.8216	0.6745	0.7703
80	0.7	0.8175	0.9591	0.7785	0.9312	0.7137	0.8722	0.6568	0.7869
80	0.8	0.799	0.9326	0.761	0.9054	0.6788	0.8164	0.6575	0.8317
80	0.9	0.7825	0.9007	0.7453	0.8745	0.6728	0.8359	0.6527	0.8047
100	0.7	0.7159	0.8807	0.6818	0.8551	0.6794	0.8571	0.6216	0.8094
100	0.8	0.7827	0.8968	0.7455	0.8707	0.6673	0.8058	0.6464	0.7764
100	0.9	0.7399	0.8936	0.7047	0.8675	0.6159	0.8282	0.6015	0.7978

9 Summary

In this study, an EAFSS problem with an exponential time-dependent learning effect is investigated. As shown in Section 4, the conveyed message is scheduling of minimizing energy consumption is a generalization of minimizing makespan for the proposed problem, and the perspective of economics and environment are highly related in this scope. For the solution, the newly proposed solution framework BBNP is found to be effective and shows a better performance in the numerical experiment. SPT is shown to have a tight bound with the worst-case analysis. Composite bounds are developed and shown to provide considerable computational advantages. However, no superiority is observed among the different LBs.

For future research, many complicated manufacturing environments, such as job shop and flexible job shop can be introduced and investigated. Multi-objective optimization can be conducted to investigate the relationship between different factors.

References

- Agrawal P, Rao S (2014). Energy-aware scheduling of distributed systems. *IEEE Transactions on Automation Science and Engineering*, 11(4): 1163–1175
- Artigues C, Lopez P, Haït A (2009). Scheduling under energy constraints. In: *International Conference on Industrial Engineering and Systems Management (IESM 2009)*
- Biskup D (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1): 173–178
- Chan S H, Lam T W, Lee L K (2013). Scheduling for weighted flow time and energy with rejection penalty. *Theoretical Computer Science*, 470: 93–104
- Cheng T E, Wang G (2000). Single machine scheduling with learning effect considerations. *Annals of Operations Research*, 98(1–4): 273–290
- Gao F, Liu M, Wang J J, Lu Y Y (2017). No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times. *International Journal of Production Research*, (2): 1–9
- Garey M R, Johnson D S (1979). *Computers and Intractability (Vol. 29)*. New York: WH Freeman
- Gonzalez T, Sahni S (1978). Preemptive scheduling of uniform processor systems. *Journal of the Association for Computing Machinery*, 25(1): 92–101
- Johnson S M (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics*, 1(1): 61–68 (NRL)
- Koulamas C, Kyparisis G J (2007). Single-machine and two-machine flowshop scheduling with general learning functions. *European Journal of Operational Research*, 178(2): 402–407
- Kuo W H, Hsu C J, Yang D L (2012). Worst-case and numerical analysis of heuristic algorithms for flowshop scheduling problems with a time-dependent learning effect. *Information Sciences*, 184(1): 282–297
- Lee W C, Wu C C (2004). Minimizing total completion time in a two-machine flowshop with a learning effect. *International Journal of Production Economics*, 88(1): 85–93
- Lee Y C, Zomaya A Y (2010). Energy efficient resource allocation in

- large scale distributed systems. In: Proceedings of Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on IEEE. 580–583
- Li G, Wang X Y, Wang J B, Sun L Y (2013). Worst case analysis of flow shop scheduling problems with a time-dependent learning effect. *International Journal of Production Economics*, 142(1): 98–104
- Liu G S, Yang H D, Cheng M B (2017). A three-stage decomposition approach for energy-aware scheduling with processing-time-dependent product quality. *International Journal of Production Research*, 55(11): 3073–3091
- Liu J, Chou P H, Bagherzadeh N, Kurdahi F (2001). Power-aware scheduling under timing constraints for mission-critical embedded systems. In: Proceedings of the 38th Annual Design Automation Conference. 840–845
- Pei J, Cheng B, Liu X, Pardalos P M, Kong M (2017). Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. *Annals of Operations Research*, (3): 1–25
- Pei J, Liu X, Fan W, Pardalos P M, Lu S (2017). A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers. *Omega*, 82: 55–69
- Rudek R (2011). Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. *Computers & Industrial Engineering*, 61(1): 20–31
- Shi L, Ólafsson S (2000). Nested partitions method for global optimization. *Operations Research*, 48(3): 390–407
- Shi L, Ólafsson S, Chen Q (2001). An optimization framework for product design. *Management Science*, 47(12): 1681–1692
- Smutnicki C (1998). Some results of the worst-case analysis for flow shop scheduling. *European Journal of Operational Research*, 109(1): 66–87
- Sun L H, Cui K, Chen J H, Wang J, He X C (2013). Research on permutation flow shop scheduling problems with general position-dependent learning effects. *Annals of Operations Research*, 211(1): 473–480
- Wang J B, Wang D, Wang L Y, Lin L, Yin N, Wang W W (2009). Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times. *Computers & Mathematics with Applications*, 57(1): 9–16
- Wang J B, Wang J J (2014). Flowshop scheduling with a general exponential learning effect. *Computers & Operations Research*, 43: 292–308
- Wang J B, Xia Z Q (2005). Flow-shop scheduling with a learning effect. *Journal of the Operational Research Society*, 56(11): 1325–1330
- Wang X Y, Zhou Z, Zhang X, Ji P, Wang J B (2013). Several flow shop scheduling problems with truncated position-based learning effect. *Computers & Operations Research*, 40(12): 2906–2929
- Webb G K (1994). Integrated circuit (IC) pricing. *Journal of High Technology Management Research*, 5(2): 247–260
- Wright T P (1936). Factors affecting the cost of airplanes. *Journal of the Aeronautical Sciences*, 3(4): 122–128
- Wu C C, Lee W C (2009). A note on the total completion time problem in a permutation flowshop with a learning effect. *European Journal of Operational Research*, 192(1): 343–347
- Wu T, Shi L, Duffie N A (2010). An HNP-MP approach for the capacitated multi-item lot sizing problem with setup times. *IEEE Transactions on Automation Science and Engineering*, 7(3): 500–511
- Xu J, Lin W C, Wu J, Cheng S R, Wang Z L, Wu C C (2016). Heuristic based genetic algorithms for the re-entrant total completion time flowshop scheduling with learning consideration. *International Journal of Computational Intelligence Systems*, 9(6): 1082–1100
- Xu Z, Sun L, Gong J (2008). Worst-case analysis for flow shop scheduling with a learning effect. *International Journal of Production Economics*, 113(2): 748–753