Omega 82 (2019) 55–69

Contents lists available at ScienceDirect

Omega

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

A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers[☆]

Jun Pei[a,b,*], Xinbao Liu[a,c,*], Wenjuan Fan[a,c], Panos M. Pardalos[a,b,*], Shaojun Lu[a,c]

[a] School of Management, Hefei University of Technology, Hefei, China
[b] Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, USA
[c] Key Laboratory of Process Optimization and Intelligent Decision-making of Ministry of Education, Hefei, China

ARTICLE INFO

Article history:
Received 25 August 2017
Accepted 7 December 2017
Available online 15 December 2017

Keywords:
Scheduling
Serial-batching
Deteriorating jobs
Financial budget
Resource constraint
Multiple manufacturers

ABSTRACT

We study a coordinated serial-batching scheduling problem that features deteriorating jobs, financial budget, resource constraints, resource-dependent processing times, setup times, and multiple manufacturers simultaneously. A unique feature but also a significant challenge in this problem is the dual constraints on resources, i.e., financial budget and resource quantity. Some key structural properties are first identified for the setting where the jobs and resources are already assigned to each manufacturer, which enables us to develop the optimal resource allocation scheme. Then, a polynomial-time scheduling rule is proposed to search for the optimal solution for each manufacturer in this setting. Then, a hybrid BA-VNS algorithm combining Bat algorithm (BA) and variable neighborhood search (VNS) is proposed to tackle the studied problem, and the optimal scheduling rule is implemented in its encoding procedure. Finally, computational experiments are conducted to test the performance of the proposed algorithm, and the efficiency and improvements are compared with those of BA, VNS, and Particle Swarm Optimization (PSO), with respect to convergence speed as well as computational stability.

1. Introduction

The batch scheduling problems exist in many practical manufacturing scenarios, which constitute a significant research area in scheduling and have captured the attention of many researchers (see [1,2]). In this paper, a coordinated serial-batching scheduling problem is studied, characterized by deteriorating jobs, resource constraints, resource-dependent processing times, setup times, and multiple manufacturers simultaneously. In particular, dual resource constraints are considered, specifically, on the quantity of the actual resources and the costs of financial resources. This problem has significant practice relevance. In a typical manufacturing industry, increasing peer competition and cost pressure bring higher requirements on production efficiency. Many enterprises operate largely with resource limits, meanwhile struggling to reduce production time. In view of this, how to effectively schedule the production process under limited resources is a problem of great interests and significance. Furthermore, considering the actual production scenario, some important features, e.g., deteriorating effect and batch processing, should be taken into account. However, to the best of our knowledge, rare study has investigated this type of problems up to now.

Financial constraint or financial budget is a common problem in many practical production scenarios (see, e.g., [3–7]). This set of problems considers the constraints on basically immaterial resources (i.e., money) consumed by actual non-renewable resources (e.g., fuel, materials, etc.) for processing jobs. These financial resources mainly contribute to improving managerial decision-making by providing more profits that can be easily calculated [8]. By extending the classical scheduling problems to the financial constraints setting, some researchers have conducted specific studies in the last decade [9,10]. On the other hand, production resource constraint is another inevitable issue that should also be considered in manufacturing process, given that resources cannot be unlimitedly increased. It should be noted here that in most cases, job processing times are related to the quantity of resources, and thus the production speed and effectiveness can be enhanced through increase in resources [11].

Thus, here comes the problem, i.e., how to find an optimal schedule of the jobs considering resource amount constraint and

☆ This manuscript was processed by Associate Editor AE Prokopyev.

* Corresponding authors at: School of Management, Hefei University of Technology, Hefei, China.

E-mail addresses: feiyijun.ufl@gmail.com (J. Pei), lxb@hfut.edu.cn (X. Liu), pardalos@ufl.edu (P.M. Pardalos).

https://doi.org/10.1016/j.omega.2017.12.003

financial budget for the resource, which are closely correlated with each other. Some related studies can be found in [12–15] etc. Specifically, recent research has achieved substantial improvements under the financial and resource allocation constraints when jobs are processed in a general way. However, more particular and practical scheduling problems considering batch processing way, known as batch scheduling problem, as well as the financial budget and resource constraint are rarely explored. In this paper, we consider these conditions simultaneously for the first time. The mixed features make our studied scheduling problem much more complicated than the general processing way setting.

The batch scheduling problems, mostly under two different processing patterns, i.e., parallel-batching and serial-batching (see, for example [16,17]), have been investigated extensively considering a variety of settings and features. Particularly, in real-world situations the job processing times always fluctuate. For example, due to the decreasing of the machine performance or other reasons, some additional time is required for processing the job, which is known as the effect of job deterioration [18,19]. The scheduling problems involving the deteriorating jobs are first proposed by Gupta and Gupta [20]. Since then, the deteriorating effect considering different jobs processing ways has aroused increasing attention. Detailed illustrations can be found in [21–24]. For solving the batching scheduling problems, some metaheuristics have been developed in recent studies. Shahvari and Logendran provide an efficient meta-heuristic algorithm based on tabu search with multi-level diversification and multi-tabu structure, which moves back and forth between batching and scheduling phases [25]. Arbib et al. propose a tabu-search algorithm exploring batch output sequences [26]. Besides, the improved genetic algorithms are also developed in some batching scheduling problems. Vallada and Ruiz present a genetic algorithm for the unrelated parallel machine scheduling problem in which machine and job sequence dependent setup times are considered [27]. Francisco and Pedro propose three genetic algorithms with different types of crossovers to solve the short-term scheduling problem of batch processes [28].

More specifically, the scheduling problems incorporating learning/deteriorating effect have already been studied, with the constraints on financial budget and actual resources' quantity [29,30]. Lately, the considerations on resource-dependent processing times, including learning/deteriorating effect, provide important insights into the properties of batch-scheduling problem [31,32]. To best deploy the scheduling practices, relevant algorithms have been developed [33,34], including some important heuristic algorithms proposed by Gafarov et al. [3], Wang and Wang [31], Zhu et al. [32], and He et al. [34]. In addition to the studies mentioned above, researchers also pay attention to the resource scheduling problems related to assignment problems (e.g., [33]). Some researchers provide polynomial time algorithms to solve the scheduling problems considering deteriorating effect. Even with the introduction of time-dependent deterioration to job processing times, the single-machine makespan minimization problem remains polynomial solvable, see [35–38]. In real production, the deteriorating effect exists when machine efficiency decreases during continuous processing.

However, there is limited existing research on practical scheduling problems combining learning/deteriorating effect, resource-dependent processing times, financial budget, and batch processing way simultaneously. Thus, this paper contributes to building a bridge between the serial-batching scheduling and practically significant constraints as well as to optimize the makespan. In the past research, we have already studied some similar problems [23,39–44]. In this paper we focus on exploring serial-batching scheduling problem associated with deteriorating jobs, financial budget, and resource constraint simultaneously, making it differentiated from our previous research. Some structural properties as well as a novel hybrid BA-VNS algorithm are proposed to solve this problem, which differs greatly from [41] and [43]. The comparisons of existing literatures and current study are shown in Table 1.

The main contributions of this paper can be summarized as follows:

(1) A coordinated serial-batching scheduling problem that features deteriorating jobs, resource constraints, resource-dependent processing times, setup times, and multiple manufacturers is investigated. Specifically, we incorporate a unique feature into this problem, which makes it much more challenging, i.e., the dual constraints of resource on financial budget and resource quantity.
(2) Under the specific situation that the jobs and resources are pre-assigned to each manufacturer, we first identify some key structural properties, based on which an optimal resource allocation scheme is developed. Then, a polynomial-time optimal scheduling rule is proposed for each manufacturer in such situation.
(3) For the general situation, an effective hybrid BA-VNS algorithm which combines Bat Algorithm (BA) and Variable Neighborhood Search (VNS) is proposed to tackle the studied problem, and the optimal scheduling rule is implemented in its encoding procedure.

The remainder of this paper is organized as follows: the notations and problem statement are illustrated in Section 2. Some structural properties and the scheduling rule of the proposed problem are derived in Section 3. In Section 4, a hybrid BA-VNS algorithm combining with the proposed optimal algorithms is developed to solve the problem, and some computational experiments are conducted to testify its performance. We conclude this paper and give some future research directions in Section 5.

2. Notations and problem statement

The notations used throughout this paper are given in Table 2.

The considered problem can be formally described as follows. There is a set of N independent and non-preemptive jobs to be processed on the serial-batching machines in q manufacturers. Each manufacturer has a single serial-batching machine, and they cooperate to finish the production task. All the jobs are first assigned and then processed in each manufacturer. The framework of the studied problem is shown in Fig. 1.

All the jobs and machines are available at time t_0, and these jobs are processed in the way of serial batches such that jobs within the same batch are processed one after another in a serial fashion [45]. The number of jobs in each batch cannot exceed the machine capacity c. We follow the resource-dependent processing time model with the deteriorating jobs as proposed by Wei et al. [46] and Wang and Wang [47], and further extend this model into the setting of serial-batching production way.

$$p_{[i]} = a_i + bt - \theta u_i, \ \ i = 1, 2, \ldots, N$$

where $a_i > 0$ is the normal processing time of J_i, $b(>0)$ is the deteriorating rate of all jobs, $\theta(>0)$ is the resource's compression rate of all jobs' processing time, and u_i is the amount of a non-renewable resource allocated to job J_i with $0 \le u_i \le \frac{a_i}{\theta}$.

Dual constraints of the resource are considered. The first constraint is that the total amount of the non-renewable manufacturing resource for all manufacturers is limited, that is, $\sum_{i=1}^{N} u_i \le U$, and the second one limits the financial budget on the non-renewable manufacturing resources for each manufacturer, i.e., $\omega U_g \le B_g$.

The batches' setup time is required before processing each batch, and it is defined as follows:

$$s_k = \mu t$$

Table 1
Comparisons of our previous work, existing literatures and current study.

Publication	Effect		Setup time	Objective	Jobs processing	Algorithm	Constraints	
	LE	DE					FC	RC
Pei et al. [23]		√	TD	C_{max}	Serial-batching	Heuristic algorithm	–	–
Pei et al. [39]		√	TD	C_{max}; $\sum_{i=1}^{n} C_i$;	Serial-batching	Heuristic algorithm	–	–
Pei et al. [40]	√	√	TD	C_{max}; $\sum_{i=1}^{n} U_i$; E_{max}	Serial-batching	Heuristic algorithm	–	–
Fan et al. [41]	√	√	–	C_{max}	Serial-batching	Heuristic algorithm and VNS-ASHLO	–	–
Pei et al. [42]		√	TD	C_{max}; $\sum_{i=1}^{n} U_i$; $\sum_{i=1}^{n} C_i$	Serial-batching	Heuristic algorithm	–	–
Pei et al. [43]	√		TD	E_{max}; $\sum_{i=1}^{n} U_i$	Serial-batching	Heuristic algorithm and VNS-GSA	–	–
Pei et al. [44]	√		TD	C_{max}	Serial-batching	Heuristic algorithm and GSA-TS	–	
Gafarov et al. [3]	–	TD	ΣT_j	General processing	Heuristic algorithm		√	
Wang and Wang [31]		√	–	d^*, u^*, $f(d, u, \pi)$	General processing	Heuristic algorithm		√
Zhu et al. [32]	√			A cost function	Group processing	Heuristic algorithm		√
Wei et al. [33]		√	–	A cost function	General processing	ASS		√
He et al. [34]	√		–	A cost function	General processing	Heuristic algorithm		√
Current study		√	TD	C_{max}	Serial-batching	Heuristic algorithm and BA-VNS	√	√

Note: C_{max} denotes makespan, $\sum_{i=1}^{n} C_i$ denotes total completion time, $\sum_{i=1}^{n} U_i$ denotes number of delayed *jobs,* E_{max} *denotes maximum earliness,* ΣT_j *denotes total tardiness, and d and u denote due date and resources amount. LE and DE denote learning effect and deteriorating effect respectively, TD denotes time-dependent setup time, and FC and RC denote financial constraint and resource constraint, respectively.*

Table 2
Notations.

Notation	Definition
N	The number of jobs
q	The number of manufacturers
J_i	Job i, $i = 1, 2, \ldots, n$
M_g	The serial-batching machine in the gth manufacturer, $g = 1, 2, \ldots, q$
N_g	The number of jobs processed on M_g, $g = 1, 2, \ldots, q$
U	The total amount of a non-renewable resource
b	The deteriorating rate of all jobs
a_i	The normal processing time of J_i, $i = 1, 2, \ldots, N$
θ	The resource's compression rate of all jobs' processing time
u_i	The amount of a non-renewable resource allocated to job J_i, $0 \le u_i \le \frac{a_i}{\theta}$, $i = 1, 2, \ldots, N$
U_g	The total amount of a non-renewable resource allocated to M_g, $g = 1, 2, \ldots, q$, i.e., $\sum_{g=1}^{q} U_g = U$.
ω	The unit cost of the non-renewable resource
$p_{[i]}$	The actual processing time of J_i, $i = 1, 2, \ldots, N$
m	The number of batches
m_g	The number of batches on M_g, $g = 1, 2, \ldots, q$
b_{gk}	The kth batch processed on M_g, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
n_{gk}	The number of jobs in b_{gk}, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
μ	The deteriorating rate of batches' setup time
s_{gk}	The setup time of b_{gk}, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
c	The capacity of the batching machine
B_g	The financial resource budget for the gth manufacturer, $g = 1, 2, \ldots, q$
$S(b_{gk})$	The starting time of b_{gk}, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
$C(b_{gk})$	The completion time of b_{gk}, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
$P(b_{gk})$	The actual processing time of b_{gk}, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$
x_{igkr}	1, if J_i is assigned in the rth position of b_{gk}; 0, otherwise, $k = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$, $r = 1, 2, \ldots, n_{gk}$
C_{max}^{j}	The maximum completion time of jobs on M_g, $g = 1, 2, \ldots, m$
C_{max}	The makespan

where μ is the deteriorating rate of batches' setup time, and t is the starting time of processing any batch.

For the studied problem, the following decisions should be made:

(i) how to assign all jobs on the serial-batching machine of each manufacturer for processing,
(ii) how to assign the resources to each jobs subject to dual constraints of the resource, i.e., the total quantity of resources and financial budget for the resource,
(iii) how to break the assigned jobs into jobs batches on each machine,
(iv) how to sequence the jobs in each batch, and finally
(v) how to sequence the batches on each machine,

with the overall objective to minimize the maximum completion time of all jobs on each machine.

In the following sections, we first investigate the structural properties of the problem given that the resources have been assigned to each manufacturer, under which circumstance we propose a scheduling rule for each manufacturer. Based on the structural properties and the scheduling rule, an effective hybrid BA-VNS algorithm which integrates Bat algorithm with Variable Neighborhood Search is developed to solve the studied problem. According to the three-field notation schema $\alpha|\beta|\gamma$ introduced by Graham et al. [48], the studied problem in this paper is denoted as $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$.

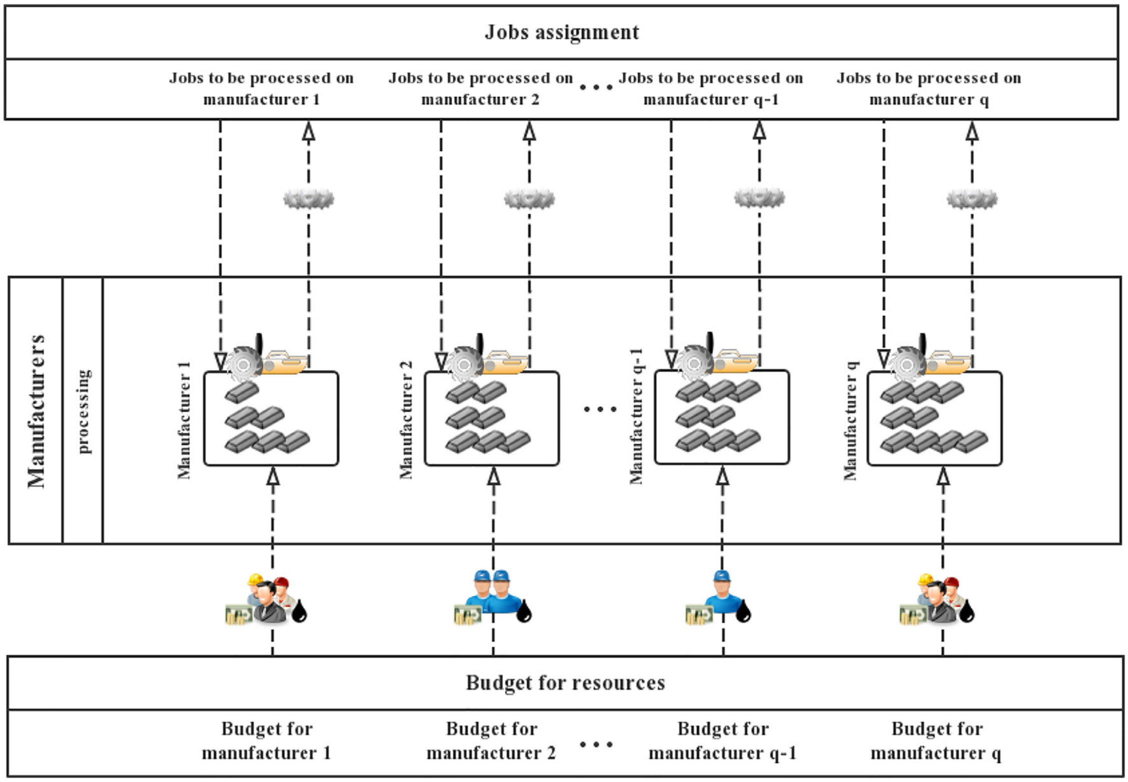


Fig. 1. The framework of the scheduling problem.

3. The structural properties and the scheduling rule

In this section, the completion time of each batch in a certain manufacturer is derived, and we further use it in the studied problem. Then, we investigate the structural properties of resource allocation, jobs batching and batches sequencing given that the resource has been assigned to each manufacturer.

Proposition 1. *For any given schedule $\pi = (b_{11}, b_{12}, \ldots, b_{lf}, \ldots, b_{qm_g})$, the completion time of b_{lf} in schedule π is*

$$C\left(b_{lf}\right) = t_0(1+\mu)^f(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f} n_{ld}-\sum_{d=1}^{k-1} n_{ld}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{ilkr} \tag{3.1}$$

where $l = 1, 2, \ldots, g,\ f = 1, 2, \ldots, m_l$.

Proof. This lemma can be proved by the mathematical induction based on the number of batches. First, for $f = 1$, we have

$$C(b_{l1}) = s_{11} + \sum_{r=1}^{n_{11}}\sum_{i=1}^{N} p_{[i]}x_{il1r} = t_0(1+\mu)(1+b)^{n_{l1}} + \sum_{r=1}^{n_{l1}}(1+b)^{n_{l1}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{i11r}.$$

Thus, Eq. (3.1) holds for $f = 1$. Suppose for all $2 \le f \le \mathrm{m}_l - 1$, Eq. (3.1) is satisfied. We have

$$C\left(b_{lf}\right) = t_0(1+\mu)^f(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f} n_{ld}-\sum_{d=1}^{k-1} n_{ld}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{ilkr}.$$

Then, for the $(f+1)th$ batch $b_{l(f+1)}$,

$$\begin{aligned}
C\left(b_{l(f+1)}\right) &= C\left(b_{lf}\right)(1+\mu) + \sum_{r=1+\sum_{k=1}^{f} n_{lk}}^{\sum_{k=1}^{f+1} n_{lk}} p_{[r]} \\
&= (1+\mu)\left[t_0(1+\mu)^f(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f} n_{ld}-\sum_{d=1}^{k-1} n_{ld}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{ilkr}\right] \\
&\quad + \sum_{r=1}^{n_{l(f+1)}}(1+b)^{\sum_{d=1}^{f+1} n_{ld}-\sum_{d=1}^{f} n_{ld}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{il(f+1)r} \\
&= t_0(1+\mu)^{f+1}(1+b)^{\sum_{k=1}^{f+1} n_{lk}} + \sum_{k=1}^{f+1}(1+\mu)^{f+1-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld}-\sum_{d=1}^{k-1} n_{ld}-r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{ilkr}.
\end{aligned}$$

Thus, Eq. (3.1) holds for $C(b_{l(f+1)})$, and the lemma is proved. □

Based on the above result of Proposition 1, we further investigate the following properties on the argument of jobs number in all batches and the number of batches.

Proposition 2. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, it should be $n_{gk} \le n_{g(k+1)}$ for all batches processed on a certain machine M_g, where $k = 1, 2, \ldots, m_g - 1,\ g = 1, 2, \ldots, q$.*

Proof. Here we assume that π_g^* and π_g are an optimal schedule and a job schedule on a certain machine M_g, respectively. The difference between these two schedules is the transferring of a job J_u from b_{lf} to $b_{l(f+1)}$ $(f = 1, 2, \ldots, m_l - 1,\ l = 1, 2, \ldots, q)$, and J_u is assigned to the last position of b_{lf} in the optimal schedule π_g^*. That is, $\pi_g^* = (W_1, b_{lf}, b_{l(f+1)}, W_2)$, $\pi_g = (W_1, (b_{lf}/\{J_u\}), (b_{l(f+1)} \cup \{J_u\}), W_2)$, where $n_{lf} \geq 2$, W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. J_u is the last one in b_{lf} and then updated to the last one in $b_{l(f+1)}$.

For π_g^*, the completion time of $b_{l(f+1)}$ is

$$C\big(b_{l(f+1)}(\pi_g^*)\big) = t_0(1+\mu)^{f+1}(1+b)^{\sum_{k=1}^{f+1} n_{lk}} + \sum_{k=1}^{f+1}(1+\mu)^{f+1-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld} - \sum_{d=1}^{k-1} n_{ld} - r}\sum_{i=1}^{N}(a_i - \theta u_i)x_{ilkr}.$$

Then, for π_g, the completion time of $b_{l(f+1)}$ is

$$C\big(b_{l(f+1)}(\pi_g)\big) = t_0(1+\mu)^{f+1}(1+b)^{\sum_{k=1}^{f+1} n_{lk}} + \sum_{k=1}^{f+1}(1+\mu)^{f+1-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld} - \sum_{d=1}^{k-1} n_{ld} - r}\sum_{i=1}^{N}(a_i - \theta u_i)x_{ilkr} - (1+\mu)(1+b)^{n_{l(f+1)}}(a_u - \theta u_u) + (1+b)^{n_{l(f+1)}} \times (a_u - \theta u_u).$$

Consequently,

$$\begin{aligned} &C\big(b_{l(f+1)}(\pi_g^*)\big) - C\big(b_{l(f+1)}(\pi_g)\big) \\ &= (1+\mu)(1+b)^{n_{l(f+1)}}(a_u - \theta u_u) - (1+b)^{n_{l(f+1)}}(a_u - \theta u_u) \\ &= \mu(1+b)^{n_{l(f+1)}}(a_u - \theta u_u). \end{aligned}$$

It can be derived that $C(b_{l(f+1)}(\pi_g^*)) > C(b_{l(f+1)}(\pi_g))$, which conflicts with the optimal solution. Thus, we can keep transferring the last job from b_{lf} to $b_{l(f+1)}$ until $n_{gk} \leq n_{g(k+1)}$. The proof is completed. □

Based on Proposition 2 and the transferring operations of jobs, the following lemma can be further obtained.

Proposition 3. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, there should be $\lceil \frac{N_g}{c} \rceil$ batches in the gth manufacturer $(g = 1, 2, \ldots, q)$ and all batches are full of jobs except possibly the first batch for the batches processed in any manufacturer in the optimal schedule.*

Next, we analyze the properties for the assigned resource for all jobs.

Proposition 4. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, there exits an optimal solution which should be $u_v \leq min\{u_u, \frac{a_v}{\theta}\}$ when J_v is processed after J_u in a certain manufacturer.*

This proof can be also completed by the transferring operations of the resource, and we omit it.

Based on the result of Proposition 4, the following property on resource allocation can be further obtained.

Proposition 5. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, if all jobs and resource have been assigned in the manufacturers, then there exits an optimal solution where $u_v = min\{U_g - \sum_{k=1}^{f-1}\sum_{r=1}^{n_{gk}}\sum_{i=1}^{N} x_{igkr}u_i - \sum_{r=1}^{d-1}\sum_{i=1}^{N} x_{igfr}u_i, \frac{a_i}{\theta}\}$ when J_v is in the dth position of b_{gf} on M_g, $f = 1, 2, \ldots, m_g$, $g = 1, 2, \ldots, q$.*

Then, combining the results of Propositions 4 and 5, we further investigate the properties of job sequencing and resource assignment in any manufacturer.

Proposition 6. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, if all jobs and resource have been assigned in the manufacturers, then there exits an optimal solution where all jobs in any manufacturer should be sequenced in the non-decreasing order of a_i $(i = 1, 2, \ldots, n)$.*

Proof. We consider any two jobs in a manufacturer and then two cases need to be considered, where one case is that these two jobs are from the same batch, and the other one is that they are from different batches. Here we consider the first case, and assume that π_g^* and π_g are an optimal schedule and a job schedule on a certain machine M_g, respectively. The difference of these two schedules is the pairwise interchange of these two jobs J_u and J_{u+1} $(u = 1, 2, \ldots, N_i - 1)$ in the same batch, that is, $\pi^* = (W_1, J_u, J_{u+1}, W_2)$, $\pi = (W_1, J_{u+1}, J_u, W_2)$, where $J_u, J_{u+1} \in b_{lf}$, J_u and J_{u+1} are in the rth and $(r+1)$th positions of b_{lf}, $n_{lf} \geq 2$, $r = 1, 2, \ldots, n_{lf} - 1$, $f = 1, 2, \ldots, m_l$, $l = 1, 2, \ldots, q$. W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. It is assumed that $a_u \geq a_{u+1}$. Then, there exist three cases, i.e., (a) $u_u = 0$, (b) $0 < u_u < \frac{a_u}{\theta}$, and (c) $u_u = \frac{a_u}{\theta}$. Based on Propostion 5, we can further derive that it should be (a) $u_u = 0$ with $u_{u+1} = 0$, (b) $0 < u_u < \frac{a_u}{\theta}$ with $u_{u+1} = 0$, and (c) $u_u = \frac{a_u}{\theta}$ with $0 \leq u_{u+1} \leq \frac{a_{u+1}}{\theta}$. Specially, for case (b), there should be two subcases, i.e. (b-1) $0 < u_u < \frac{a_u}{\theta}$ with $u_u \leq \frac{a_{u+1}}{\theta}$ and (b-2) $0 < u_u < \frac{a_u}{\theta}$ with $u_u > \frac{a_{u+1}}{\theta}$. We first investigate subcase (b-1).

For π_g^*, the completion time of b_{lf} is

$$C\big(b_{lf}(\pi_g^*)\big) = t_0(1+\mu)^{f}(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f+1-k} \times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld} - \sum_{d=1}^{k-1} n_{ld} - r}\sum_{i=1}^{N}(a_i - \theta u_i)x_{ilkr}.$$

Based on Proposition 5, after the pairwise interchange of J_u and J_{u+1} in π_g, the resource assigned on J_u should be transferred to J_{u+1}. Then, the updated resource for J_{u+1} is $u'_{u+1} = u_u$, and the updated resource for J_u is $u'_u = 0$.

Then, for π_g, the completion time of b_{lf} is

$$\begin{aligned} C\big(b_{lf}(\pi_g)\big) &= t_0(1+\mu)^{f}(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f+1-k} \\ &\quad\times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld} - \sum_{d=1}^{k-1} n_{ld} - r}\sum_{i=1}^{N}(a_i - \theta u_i)x_{ilkr} \\ &\quad- \Big[(1+b)^{n_{lf}-r}\Big](a_u - \theta u_u) - \Big[(1+b)^{n_{lf}-r-1}\Big]a_{u+1} \\ &\quad+ \Big[(1+b)^{n_{lf}-r}\Big](a_{u+1} - \theta u_u) + \Big[(1+b)^{n_{lf}-r-1}\Big]a_u. \end{aligned}$$

Furthermore,

$$\begin{aligned} &C\big(b_{lf}(\pi_g^*)\big) - C\big(b_{lf}(\pi_g)\big) \\ &= \Big[(1+b)^{n_{lf}-r}\Big](a_u - \theta u_u) + \Big[(1+b)^{n_{lf}-r-1}\Big]a_{u+1} \\ &\quad - \Big[(1+b)^{n_{lf}-r}\Big](a_{u+1} - \theta u_u)\Big[(1+b)^{n_{lf}-r-1}\Big]a_u \\ &= \Big[(1+b)^{n_{lf}-r} - (1+b)^{n_{lf}-r-1}\Big][(a_u - a_{u+1})] > 0. \end{aligned}$$

It can be derived that $C(b_{lf}(\pi_g^*)) > C(b_{lf}(\pi_g))$, which conflicts with the optimal solution. Thus, J_u should be processed after J_{u+1} in subcase (b-1).

Furthermore, we investigate subcase (b-2).

The same result of $C(b_{lf}(\pi_g^*))$ can be also obtained as subcase (b-1). Similarly, based on Proposition 5, after the pairwise interchange of J_u and J_{u+1} in π_g, the resource assigned on J_u should be transferred to J_{u+1}. Then, the updated resource for J_{u+1} is $u'_{u+1} = \frac{a_{u+1}}{\theta}$, and the updated resource for J_u is $u'_u = u_u - u'_{u+1} = u_u - \frac{a_{u+1}}{\theta}$.

Then, for π_g, the completion time of b_{lf} is

$$\begin{aligned} C\big(b_{lf}(\pi_g)\big) &= t_0(1+\mu)^f(1+b)^{\sum_{k=1}^{f} n_{lk}} + \sum_{k=1}^{f}(1+\mu)^{f+1-k} \\ &\times \sum_{r=1}^{n_{lk}}(1+b)^{\sum_{d=1}^{f+1} n_{ld} - \sum_{d=1}^{k-1} n_{ld} - r}\sum_{i=1}^{N}(a_i-\theta u_i)x_{ilkr} \\ &- \left[(1+b)^{n_{lf}-r}\right](a_u-\theta u_u) - \left[(1+b)^{n_{lf}-r-1}\right]a_{u+1} \\ &+ \left[(1+b)^{n_{lf}-r}\right]\left(a_{u+1}-\theta\cdot\frac{a_{u+1}}{\theta}\right) \\ &+ \left[(1+b)^{n_{lf}-r-1}\right]\left[a_u-\theta\left(u_u-\frac{a_{u+1}}{\theta}\right)\right]. \end{aligned}$$

Furthermore,

$$\begin{aligned} & C\big(b_{lf}(\pi_g^*)\big) - C\big(b_{lf}(\pi_g)\big) \\ &= \left[(1+b)^{n_{lf}-r}\right](a_u-\theta u_u) + \left[(1+b)^{n_{lf}-r-1}\right]a_{u+1} \\ &\quad - \left[(1+b)^{n_{lf}-r}\right]\left(a_{u+1}-\theta\cdot\frac{a_{u+1}}{\theta}\right) \\ &\quad - \left[(1+b)^{n_{lf}-r-1}\right]\left[a_u-\theta\left(u_u-\frac{a_{u+1}}{\theta}\right)\right] \\ &= \left[(1+b)^{n_{lf}-r}\right](a_u-\theta u_u) + \left[(1+b)^{n_{lf}-r-1}\right]a_{u+1} \\ &\quad - \left[(1+b)^{n_{lf}-r-1}\right](a_u-\theta u_u+a_{u+1}) \\ &= \left[(1+b)^{n_{lf}-r}-(1+b)^{n_{lf}-r-1}\right](a_u-\theta u_u) > 0. \end{aligned}$$

It can be derived that $C(b_{lf}(\pi_g^*)) > C(b_{lf}(\pi_g))$, which conflicts with the optimal solution. Thus, J_u should be processed after J_{u+1} in subcase (b-2).

Thus, combining subcases (b-1) and (b-2), we can conclude that the result holds for case (b).

Then, for cases (a) and (c), the same result can be obtained in the similar way. Thus, the proof is completed. □

Propositions 2–6 establish the optimal policy for the jobs sequencing, jobs batching, batches sequencing, and resource allocation given that all jobs and resource have been assigned to a certain manufacturer. Then, we develop the following scheduling rule for each manufacturer under such circumstance.

Scheduling Rule 1:

1. Set $g = 0$.
2. Set $g = g + 1$. If $g > q$, go to step 11; Otherwise, go to step 3.
3. All jobs assigned on M_g are indexed in the non-decreasing order of a_i such that $a_1 \le a_2 \le \cdots \le a_{N_g}$, and a job list is obtained.
4. Place the first $N_g - (\frac{N_g}{c} - 1)c$ jobs in a batch.
5. If there are unscheduled jobs in the job list, then place the first c jobs in a batch and iterate. The batches are scheduled in their generation order.
6. Set $f = 0$.
7. Set $f = f + 1$ and $r = 0$. If $f > \frac{N_g}{c}$, go to step 2; Otherwise, go to step 8.
8. Set $r = r + 1$. If $r > n_{gf}$, go to step 7; Otherwise, go to step 9.
9. Set $u_r = min\{U_g, \frac{\sum_{i=1}^{N} x_{igfr} a_i}{\theta}\}$, and assign u_r resource to the job in the rth position in b_{gf}.
10. Set $U_g = U_g - u_r$. If $U_g = 0$, go to step 2; Otherwise, go to step 8.
11. Output the solutions of job scheduling and resource allocation for all manufacturers.

Theorem 1. *For the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, if all jobs and resource have been assigned for each manufacturer, then an optimal schedule can be obtained by Scheduling Rule 1 in $O(N\log N)$ time.*

Proof. Based on Propositions 1–6, an optimal solution can be generated by Scheduling Rule 1. The time complexity of steps 1 and 2 is $O(q)$, and the time complexity of steps 2–11 is at most $O(N\log N)$. Thus, the total time complexity of Scheduling Rule 1 is $O(N\log N)$. The proof is completed. □

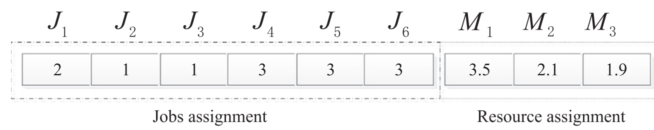


Fig. 2. Coding representation.

4. The BA-VNS algorithm

In this section, we first analyze the complexity of the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$, followed by a detailed description about the developed BA-VNS in Section 4.1, and a new local search operator is applied so as to introduce diversification into the population. In Section 4.2, we report and analyze the results of the computational experiments.

When $\theta = 0$, $b = 0$, and the machine capacity $c = 1$, the problem $P|s-batch, p_{[i]} = a_i + bt - \theta u_i, FB|C_{max}$ can be reduce to the classical parallel machines scheduling problem $P_m||C_{max}$, which is proved to be NP-hard in Garey and Johnson [49]. Then, it can be inferred that the studied problem is also NP-hard. For solving this problem in a reasonable time, we put forward a BA-VNS algorithm combining the proposed optimal algorithms. In addition, some computational experiments are conducted to test the performance of the proposed algorithm.

Variable neighborhood search (VNS) has a long record of good results obtained with hybrid methods since it was first proposed by Hansen et al. [50]. For instance, Jarboui et al. [51] develop a hybrid GA and VNS to solve no-wait flowshop scheduling problems. Kovačević et al. [52] propose a self-adaptive Differential Evolution combining VNS for continuous global optimization, and representative benchmark functions are used to verify the effectiveness of the hybrid method. Amaldass et al. [53] propose a hybrid algorithm based on Ant Colony Optimization and Variable Neighborhood Search, and this algorithm is proved effectively to solve this scheduling problem to minimize the completion time. Djenic et al. [54] consider the Bus Terminal Location Problem (BTLP) which incorporates characteristics of both the p-median and maximal covering problems, and a parallel variable neighborhood search algorithm (PVNS) is presented to solve this problem. Hansen et al. [55] and Hanafi et al. [56] explain the concept of VNS in detail and propose several new VNS based 0–1 MIP Heuristics. It is known that the key to success for VNS is the neighborhood structure applied in the algorithm. Here we put forward a new neighborhood structure based on the characteristics of our studied problem and introduce another effective metaheuristic called Bat algorithm into VNS to integrate the advantages of both algorithms. Bat algorithm (BA) is first proposed by Yang [57] inspired by bat flight. In recent years, BA has shown competitive performance in many optimization problems. Marichelvam et al. [58] solve hybrid flow shop scheduling problems using BA. Osaba et al. [59] propose an improved discrete BA for symmetric and asymmetric traveling salesman problems. Extensive experiments in these studies indicate that BA is a very effective and robust method for solving optimization problems. In this research, we design a hybrid method of BA and VNS to solve the proposed combinational optimization problem.

4.1. Key procedures of BA-VNS

4.1.1. Encoding scheme

To code solution vectors, we should consider the problem with the following three stages: (i) assign all jobs and resource to manufacturers, (ii) group the assigned jobs into batches on each machine, sequence the jobs in each batch and sequence batches on each machine, (iii) assign the resource to each job subject to the

Encoding correction strategy:

```
For position d = 1:N
    x^{d'} = Round(x^d)
    If x^{d'} < 1 then
        x^{d'} = 1
    End if
    If x^{d'} > q then
        x^{d'} = q
    End if
End for
Set sum = 0
For position d = N + 1: N + q
    If x^{d'} < 0 then
        x^{d'} = 0
    End if
    If x^{d'} > B_g/ω then
        x^{d'} = B_g/ω
    End if
    sum = sum + x^{d'}
    If sum > U then
        x^{d'} = x^{d'} − sum + U
        While(d < N + q) do
            d = d + 1
            x^{d'} = 0
        End while
        Break
    End if
End for
```

Fig. 3. Description of encoding correction strategy.

	J_1	J_2	J_3	J_4	J_5	J_6	M_1	M_2	M_3
Before correction	2.3	0.7	0.1	2.9	3.3	3.4	3.5	3.2	1.9
After correction	2	1	1	3	3	3	1.5	2.5	1

Fig. 4. An example of encoding correction strategy.

constraints of the total quantity of resource and financial resource budget. By using Scheduling Rule 1, we can further solve the problem in the second and third stages. Consequently, for the first stage, a solution to the problem of assigning jobs and resources to the manufacturers is an array of which the length is equal to the

Local search operator:

Initial constant L
Set integer $y = Rand(1, N + q)$
If $y \leq N$ then
  Set $z = Rand(1, N)$
  Set $x^{y'} = x^{z}$ and $x^{z'} = x^{y}$ and obtain the neighbor solution.
End if
If $y > N$ then
  Set $z = Rand(N + 1, N + q)$
  Set $x^{y'} = x^{y} + L * Rand(0,1)$ and $x^{z'} = x^{z} - L * Rand(0,1)$
End if

Fig. 5. Description of local search operator.

Table 3
Parameters setting.

Notation	Definition	
N	The number of jobs	20,40,60 (for small scale) 100,150,200 (for large scale)
q	The number of manufacturers	3,6,9
b	The deteriorating rate of all jobs	0.02
a_i	The normal processing time of J_i, $i = 1, 2, \ldots, N$	[3,6]
θ	The resource's compression rate of all jobs' processing time	1
ω	The unit cost of the non-renewable resource	1
μ	The deteriorating rate of batches' setup times	0.02
c	The capacity of the batching machine	3
B_g	The financial resource budget for the gth manufacturer, $g = 1, 2, \ldots, q$	[3,6]
U	Total resource limit	[25,35]

Table 4
Computational results for algorithms BA-VNS, BA, VNS, and PSO.

			BA-VNS		BA		VNS		PSO	
No.	n	M	Ave(Min)	RPD	Ave(Min)	RPD	Ave(Min)	RPD	Ave(Min)	RPD
1	20	3	25.2(24.9)	1.2	26.5(25.2)	6.5	26.1(25.2)	4.6	25.7(25.0)	3.3
2	20	6	12.7(11.6)	9.3	13.4 (12.8)	15.3	13.5(12.3)	16.4	13.0(12.2)	11.6
3	20	9	8.5(7.7)	10.3	8.7(8.1)	12.5	8.8(8.4)	14.2	8.7(7.9)	12.1
4	40	3	57.7(57.2)	0.9	60.0(57.8)	5.0	59.4(57.8)	3.9	58.6(57.7)	2.4
5	40	6	27.5(26.1)	5.6	29.3(27.1)	12.3	29.4(27.7)	13.2	27.8(26.8)	6.5
6	40	9	18.9(18.2)	3.9	21.1(20.0)	16.3	20.1(18.5)	10.5	20.1(19.4)	10.9
7	60	3	96.3(95.5)	0.8	96.7(95.9)	1.3	96.5(95.8)	1.0	97.9(96.2)	2.5
8	60	6	44.3(42.2)	5.1	45.9(44.8)	8.8	46.0(45.1)	8.9	45.1(43.6)	7.0
9	60	9	27.7(26.4)	5.0	30.1(28.5)	14.1	30.0(28.5)	13.8	28.7(27.2)	9.0
10	100	3	200.0(198.8)	0.6	205.5(203.0)	3.4	206.2(201.0)	3.7	203.6(199.3)	2.4
11	100	6	85.3(82.5)	3.4	87.8(83.4)	6.4	87.2(83.2)	5.7	89.1(84.0)	8.0
12	100	9	53.5(51.0)	4.8	54.4(52.4)	6.6	55.1(52.3)	8.0	56.5(53.3)	10.6
13	150	3	368.8(367.7)	0.3	371.5(370.5)	1.0	371.6(369.5)	1.1	372.4(369.0)	1.3
14	150	6	129.5(128.1)	1.1	141.9(137.2)	10.8	140.0(134.2)	9.1	140.0(133.4)	9.3
15	150	9	83.8(81.9)	2.3	86.7(83.7)	5.9	86.9(85.5)	6.1	87.1(83.9)	6.4
16	200	3	649.0(648.7)	0.1	651.3(649.9)	0.4	650.9(649.3)	0.4	660.1(652.1)	1.8
17	200	6	199.7(198.0)	0.9	206.3(202.1)	4.2	206.1(202.2)	4.1	210.3(205.2)	6.2
18	200	9	117.4(112.8)	4.1	124.1(119.4)	10.1	124.5(119.4)	10.4	125.2(118.6)	11.0

total number of the jobs and manufacturers. The first N position values represent the manufacturer to which the job is assigned while the last q position values represent the quantity of resources assigned to the corresponding manufacturer. It should be noted that this coding scheme is for the first time used in scheduling problems with multiple manufacturers and limited resources consideration, which is an innovative coding way to this kind of problems. Suppose that there exist six jobs and three manufacturers. A feasible object structure is presented in Fig. 2. The object suggests that jobs {2, 3}, {1}, and {4, 5, 6} with 3.5, 2.1, and 1.9 units of the resource are assigned to manufacturers 1, 2, and 3 respectively. The coding of resources and jobs is usually independent in existing literature. However, they would be too complex and difficult to be implemented in many other scheduling problems. We combine the jobs and resources in the encoding procedure, which makes the searching process simpler and more efficient. Thus, metaheuristics

BA-VNS algorithm

1. Initialize parameters, including $tmax$, $poplong$, $f_{min} = 0$, $f_{max} = [0.5,0.7,0.9,1.1]$, $E = 1$, A, r_0, α and γ
2. Set and $it = 0$, $gbest = a\ large\ enough\ positive\ constant$, randomly generate a population $pop = \{X_1, \cdots, X_l, \cdots X_{poplong}\}$ with $poplong$ bats, and each solution includes $N + q$ elements, $X_l = \{x_{l1}, \cdots, x_{ld}, \cdots, x_{lN}, \ldots, x_{l(N+q)}\}$
3. Encoding correction
4. While $(it \le tmax)$
5. Calculate fitness for each bat $popfit = \{fit_1, \cdots, fit_l, \cdots fit_{poplong}\}$.
6. For population pop, $l = 1$ to $poplong$
7. Calculate $f_l = f_{min} + (f_{max}(E) - f_{min}) * Rand(0,1)$
8. Set $V_l^t = V_l^{t-1} + (X_l^t - X_{best}) * f_l$
9. Set $X_l^t = X_l^{t-1} + V_l^t$
10. End for
11. Encoding correction
12. If $Rand(0,1) > r$ then
13. Calculate fitness for each bat $popfit = \{fit_1, \cdots, fit_l, \cdots fit_{poplong}\}$
14. Set $X_{new} = X_l$, $l = Argmin_{1<l<poplong}(fit_l)$
15. Set $X_{new} = X_{new} + \varepsilon A$
16. Encoding correction
17. End if
18. If $min\{fit_{new}, fit(X_l^t)\} < fit(X_l^{t-1})$ && $Rand(0,1) < A$, then
19. Choose the better one from X_{new} and X_l^t to update X_l
20. End if
21. If $gbest > \min\limits_{1 \le l \le poplong} fit_l$, then
22. Set $gbest = \min\limits_{1 \le l \le poplong} fit_l$, $E = 1$, and $X_{best} = X_l$
23. else
24. Set $E = E + 1$ and execute Local search operator for X_l
25. If $E > 4$ then
26. $E = 1$
27. End if
28. End if
29. Set $A = \alpha A$ and $r = r_0[1 - exp(-\gamma * it)]$
30. End while
31. Output $gbest$

Fig. 6. Pseudocode of the BA-VNS algorithm.

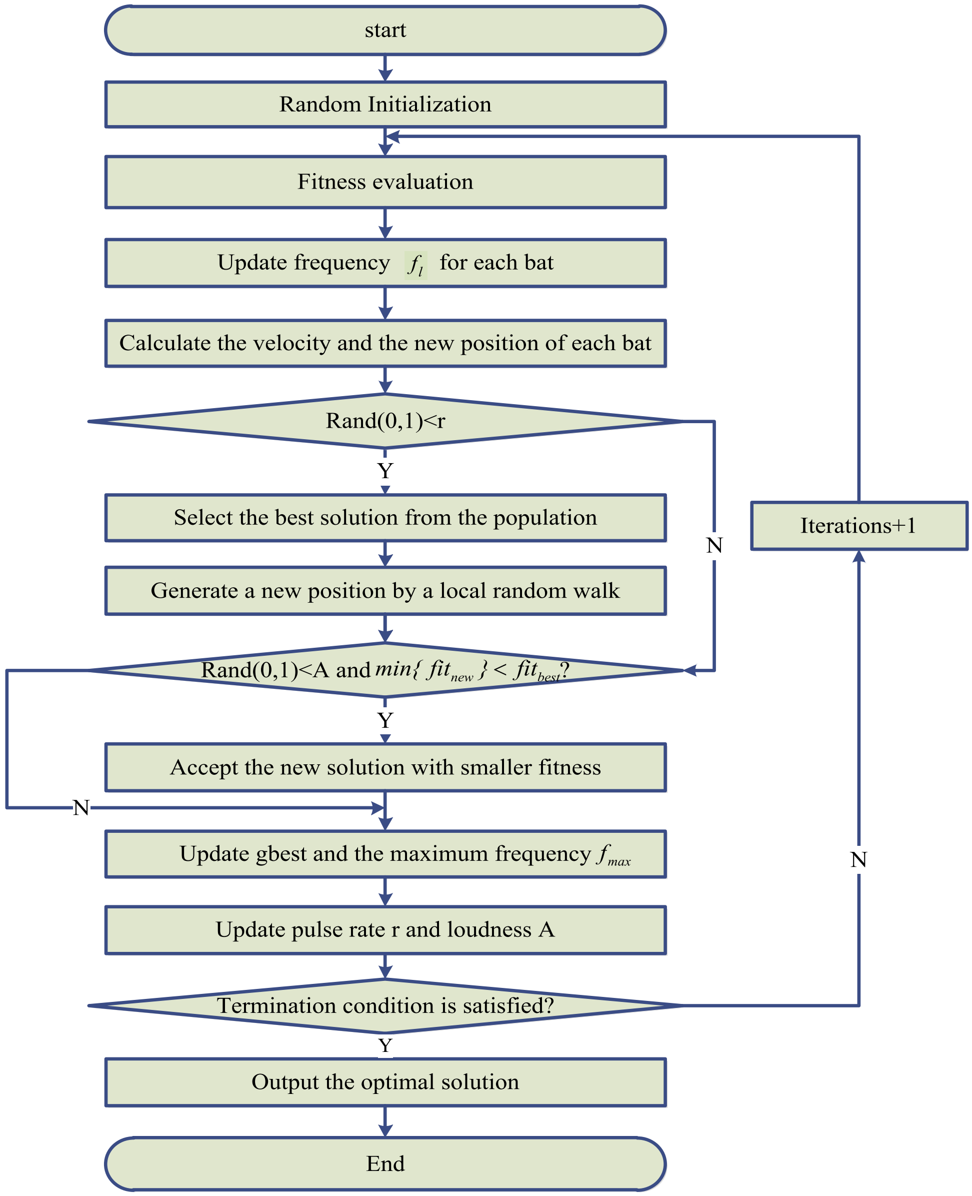


Fig. 7. Flow chart of the BA-VNS algorithm.

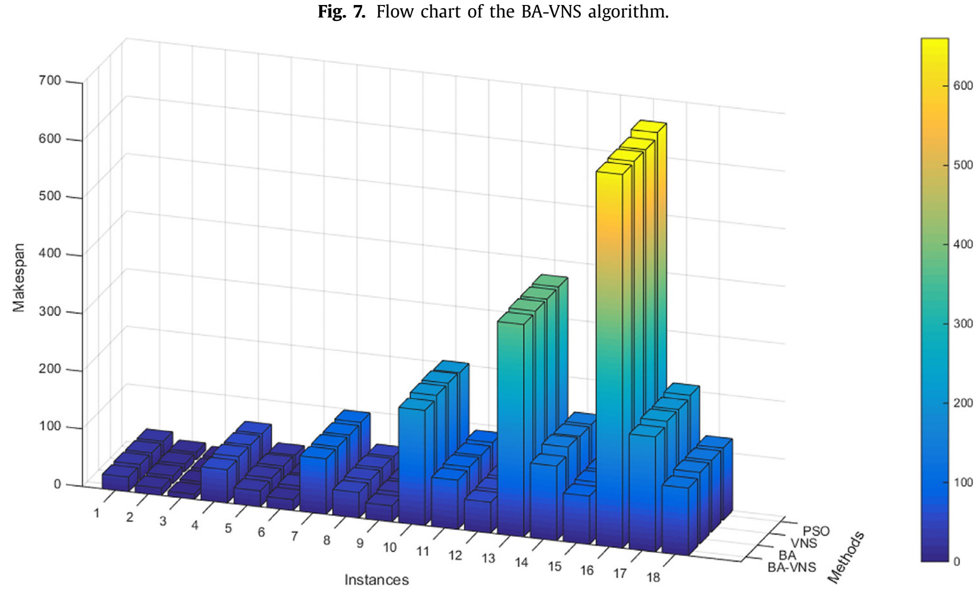


Fig. 8. Makespan in different numerical examples.

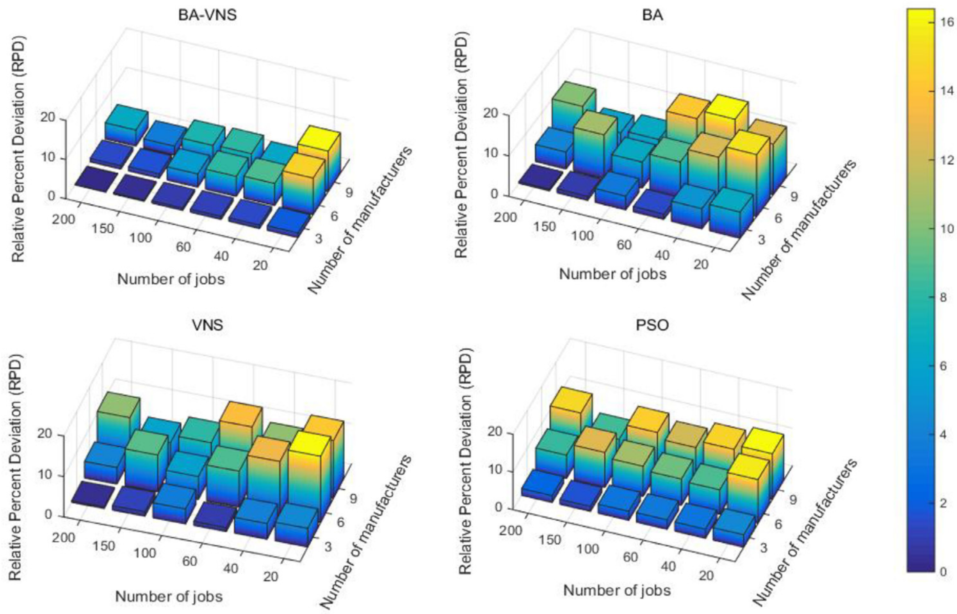


Fig. 9. RPD results with the different number of jobs and manufacturers.

can search more solutions in a shorter period of time, in turn, better solutions can be obtained. In Section 3, some key structural properties are identified, based on which an optimal resource allocation scheme is developed. Then, a polynomial-time optimal scheduling rule is proposed for each manufacturer. The highlight of the decoding is the application of the optimal scheduling rule, which greatly improves the solution quality. In addition, the decoding ensures that the schedule on each machine is optimal, and it is of great significance for real production.

4.1.2. Encoding correction strategy

In iterative processes, infeasible solutions may be generated for two reasons. Firstly, jobs should be encoded with integers while search operators may yield decimals. Similar to our previous work in [60], this paper utilizes the coding correction strategy which takes approximate values for the first N position values. Secondly, considering the constraints on financial budgets and resources, the coding correction strategy for the last q position values is designed to fix possible illegal coding during iteration process. The complete coding correction strategy is described in Fig. 3.

In order to illustrate the correction of position sequence, we give an example of an object position sequence with six jobs assigned on the machines of three manufacturers, which is shown in Fig. 4. It should be noted that $\Sigma U_g \leq 5$ and $\frac{B_g}{\omega} = \{1.5, 2.5, 2.3\}$ in the presented object.

4.1.3. Local search

Due to the uniqueness of the coding scheme in this paper, classic local search operators may be inefficient in solving the presented problem. Thus, in the proposed BA-VNS, a new local search operator is designed to improve the solution quality in each iteration, and it is shown in Fig. 5.

4.1.4. Framework of BA-VNS

In classical BA, frequencies f_{min} and f_{max} are identified in advance and remain unchanged throughout the search interaction. In our proposed BA-VNS, f_{max} is flexible and adapted to varying solutions. Actually, the hybrid method not only simulates the bat flight, but also reflects the basic thought of VNS, and its core idea is to possess the advantages of both BA and VNS. The algorithm framework of BA-VNS is described in Fig. 6. The flow chart of BA-VNS is given in Fig. 7. In the proposed hybrid BA-VNS, the neighborhoods of VNS consist of BA operators. As shown in pseudocode of the BA-VNS, f_{max} is the most important parameter in BA. However, different f_{max} values may result in varying performance of the BA in different experiments and it is hard to find a fixed f_{max} value which applies for all problem categories. Thus, we define BA with a certain $f_{\max}$ value as a neighborhood for VNS, and there are four neighborhoods for BA-VNS in this paper, that is, BA($f_{max} = 0.5$), BA($f_{max} = 0.7$), BA($f_{max} = 0.9$), and BA($f_{max} = 1.1$).

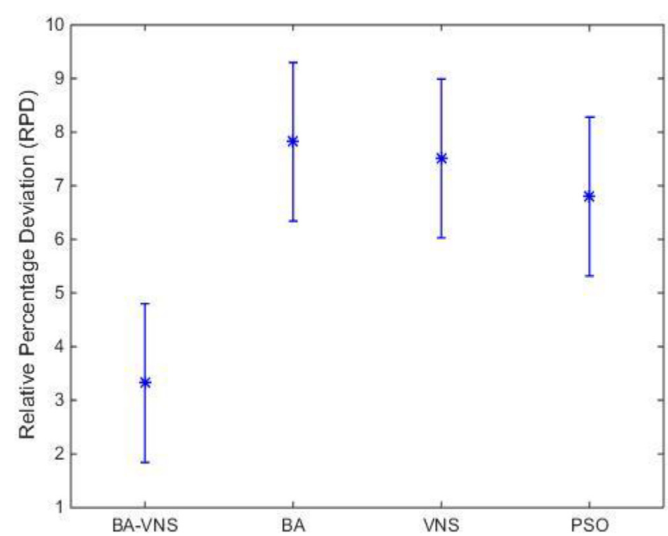
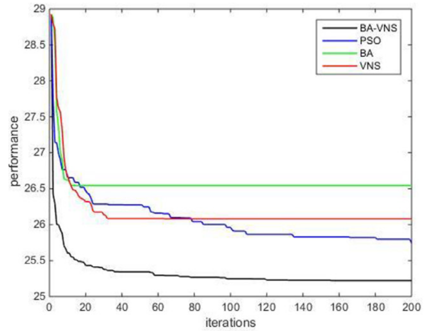


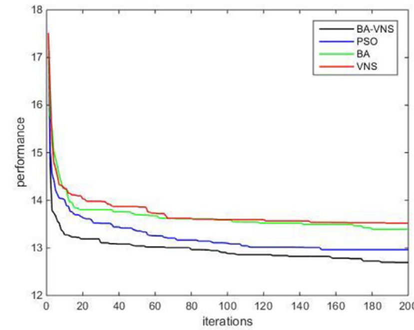
Fig. 10. Means plot and LSD intervals at the 95% confidence level for evaluated methods.

4.2. Computational experiments and comparison

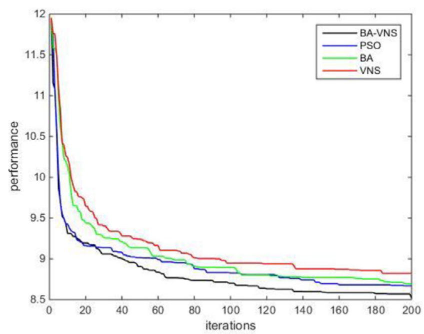
In this section, we conduct computational experiments to evaluate the performance of our proposed algorithm BA-VNS, with three classic algorithms, that is, BA [61], VNS [50], and PSO [62]. The test problems are randomly generated based on the real production as illustrated in Table 3. Based on the number of machines and jobs, 18 instances are generated in our computational experiments.



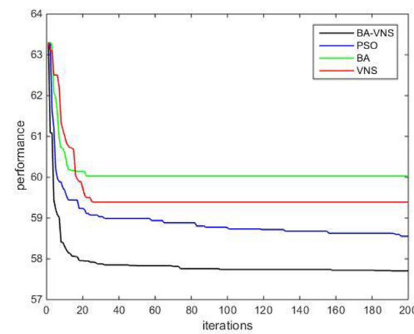
(a) Convergence curves for (20,3)



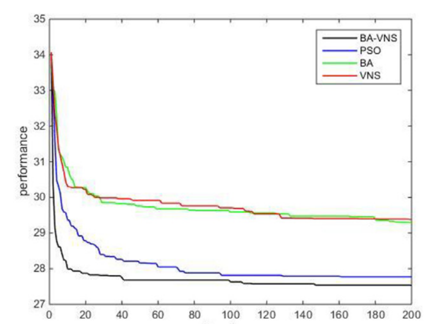
(b) Convergence curves for (20,6)



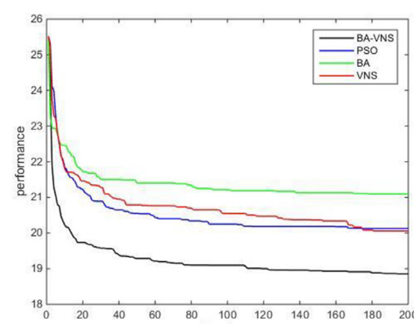
(c) Convergence curves for (20,9)



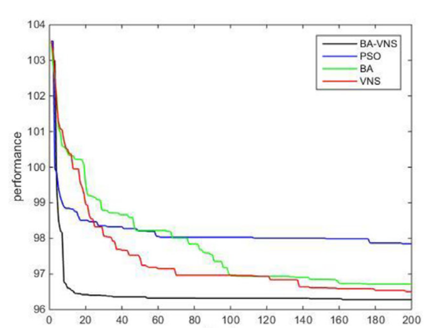
(d) Convergence curves for (40,3)



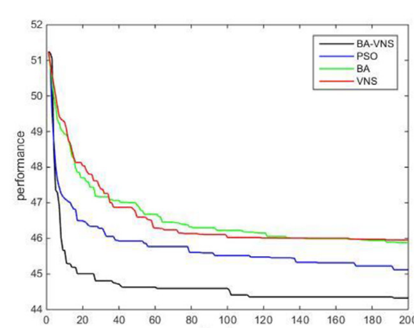
(e) Convergence curves for (40,6)



(f) Convergence curves for (40,9)

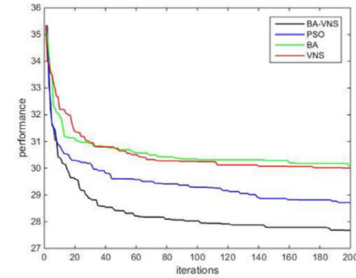


(g) Convergence curves for (60,3)

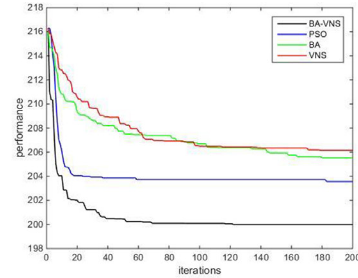


(h) Convergence curves for (60,6)

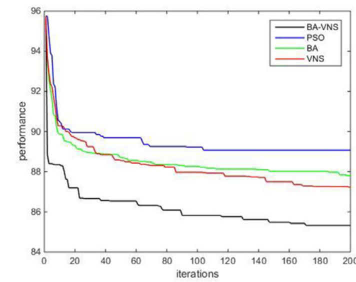
Fig. 11. Convergence behaviors of BA-VNS, BA, VNS, and PSO for each instance.



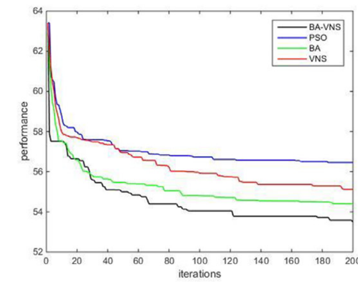
(j) Convergence curves for (60,9)



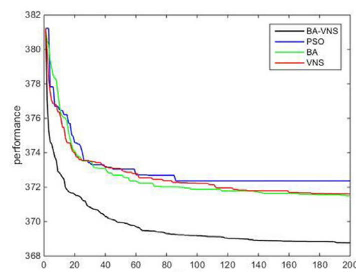
(k) Convergence curves for (100,3)



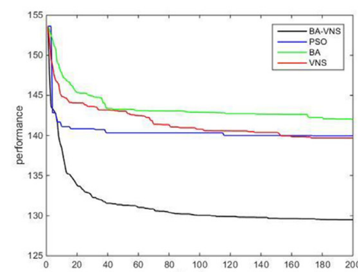
(l) Convergence curves for (100,6)



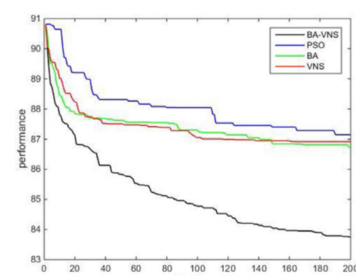
(m) Convergence curves for (100,9)



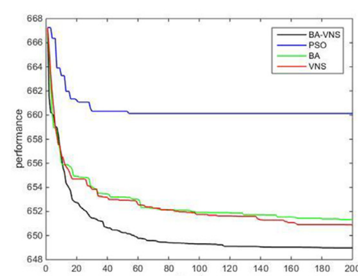
(n) Convergence curves for (150,3)



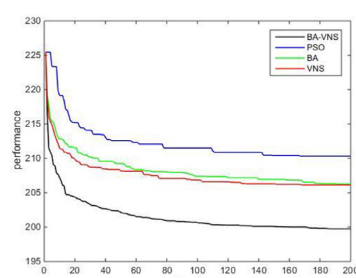
(o) Convergence curves for (150,6)



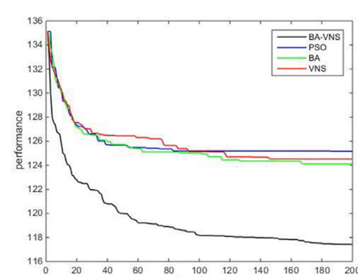
(p) Convergence curves for (150,9)



(q) Convergence curves for (200,3)



(r) Convergence curves for (200,6)



(s) Convergence curves for (200,9)

Fig. 11. Continued

We compare the results obtained by these four algorithms, and the Relative Percent Deviation (RPD) with the best-known solutions is measured for all test instances. The RPD is defined as follows:

$$RPD(ALG) = \frac{Efit(ALG) - bestfit}{bestfit} * 100 \tag{4.1}$$

where *Efit*(*ALG*) denotes the expected makespan acquired by algorithm *ALG*, and *bestfit* is the best-known value of makespan, obtained after all experiments performed. Since the meta problem is a minimization problem, a positive *RPD* indicates that the *ALG* algorithm does not find the best solution in a certain experiment. The average objective value (Avg) and the minimum objective value (Min) are measured for each problem. To make sure that algorithms can converge to a good solution, population size is set to 20, and the maximum iteration is 200. In BA-VNS, parameters are set as in [61], where $A = 1$, $r_0 = 1$, $\alpha = 0.9$, and $\gamma = 0.9$. All algorithms are implemented in C++ and run on a Lenovo computer running Window 10 with a dual-core CPU Intel i3-3240@3.40 GHz and 4 GB RAM.

Table 4 reports the average objective value, the maximum objective value, and the RPD of BA-VNS, BA, VNS, and PSO over 18 instances, where each instance is run for 20 times. From Table 4, we find that BA-VNS obtain the best average makespan, minimum makespan and RPD among the four algorithms. For instances 13–17, the RPD values of BA-VNS are even no more than 2.3, which indicates that BA-VNS is more stable as the experimental scale increases.

Figs. 8 and 9 give a more intuitive way to reveal the solution quality in graphics. From Fig. 8, the makespan increases significantly when either the number of jobs increases or the number of manufacturer decreases, and BA-VNS can continuously obtain the best solution among the four algorithms. However, it can be seen from Fig. 9 that the RPD value decreases when either the number of jobs increases or the number of manufacturers decreases, which indicates that the proposed method has a much better performance in large scale experiments. In addition, we can see that BA-VNS results in a smaller and more stable RPD value in Fig. 9. Practically, for $m = 3$, the RPD values of BA-VNS are very small, which indicates that BA-VNS can usually converge to the approximately optimal solution. Though, PSO cannot converge to a good solution while VNS and BA are unstable. From Figs. 8 and 9, we can infer that BA-VNS is more robust and efficient than other selected algorithms.

In order to show that the differences in the RPD values are statistically significant, the means plot and LSD intervals (at the 95% confidence level) for four algorithms are shown in Fig. 10. We can see that despite of the differences observed in Fig. 9, BA, VNS, and PSO are not statistically different, of which the confidence intervals are overlapped. It indicates that the performance of BA, VNS, and PSO actually performs at the same level. Furthermore, since confidence interval of BA-VNS is not overlapped with any other one and it can obtain the minimum average RPD, we can state that BA-VNS performs much better than other three algorithms from Fig. 10. This result supports the mean difference obtained from Table 4 and Fig. 9.

Fig. 11 shows the convergence behaviors of BA-VNS, BA, VNS, and PSO for each instance. In each figure, 20 tests are conducted and the average of the best solution in each iteration is listed. For small scale cases, where $n = 20, 40, 60$, PSO is usually convergent to a better solution than BA and VNS. However, for large scale cases, PSO cannot avoid trapping in local optimum and usually obtain a worse solution than BA and VNS. Compared with BA, VNS, and PSO, the hybrid BA-VNS has both faster convergence speed and better results when solving the different scales of problems. Particularly, the convergence graphs show that our hybrid BA-VNS has strong exploration ability in large-scale instances. When $n = 150, 200$, BA, VNS, and PSO all converge to a local optimal solution after 100 iterations. Though, BA-VNS can skip the local optimum and continue to find better solutions. It is worthy of mention that all the available results searched by BA-VNS are obtained within reasonable time. For example, the running time is between 20 and 25 s in the instance with 100 jobs and 6 manufacturers. Thus, we can infer that the running time of BA-VNS does not exceed 1.5 s in a single run. Based on the above experimental results, we can infer that the proposed BA-VNS is effective and robust in terms of both solution quality and convergence speed.

5. Conclusion

In this work, we study the coordinated serial-batching scheduling problem with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers, and the objective is to minimize the makespan. We first investigate the situation where the jobs and resources are already assigned to each manufacturer, and the structural properties are developed to obtain the optimal resource allocation scheme subject to dual constraints of resource. Based on the structural properties and resource allocation scheme, an optimal scheduling rule is proposed for each manufacturer in this situation. Then, a hybrid BA-VNS algorithm combining Bat algorithm and variable neighborhood search is proposed, and a comparison of the proposed algorithms with BA, VNS, and PSO shows great improvements, with respect to convergence speed as well as computational stability.

In the future research, we can extend the problem to settings with stochastic production task, which means that the processing time and required resource may be stochastic. Also, more practical constraints can be considered, such as different job sizes, dynamic job arrival, and transportation constraints. A number of interesting and important issues remain open for further research. We may also consider the cooperation of different production stages. Specifically, the multiple-agent scheduling problem with resources limits and deteriorating effect effect is worth being investigated.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 71601065, 71231004, 71501058, 71690235 and 71690230), and Innovative Research Groups of the National Natural Science Foundation of China (71521001), the Humanities and Social Sciences Foundation of the Chinese Ministry of Education (No. 15YJC630097), Anhui Province Natural Science Foundation (No. 1608085QG167). Panos M. Pardalos is partially supported by the project of "Distinguished International Professor by the Chinese Ministry of Education" (MS2014HFGY026).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.omega.2017.12.003.

References

[1] Shabtay D. The single machine serial batch scheduling problem with rejection to minimize total completion time and total rejection cost. Eur J Oper Res 2014;233(1):64–74.
[2] Li S, Ng CT, Cheng TE, Yuan J. Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. Eur J Oper Res 2011;210(3):482–8.
[3] Gafarov ER, Lazarev AA, Werner F. Single machine scheduling problems with financial resource constraints: some complexity results and properties. Math Soc Sci 2011;62(1):7–13.
[4] Tan Y, Carrillo J. Strategic analysis of the agency model for digital goods. Prod Oper Manag 2017. doi:10.1111/poms.12595.

[5] Kumar A, Tan Y. Demand effects of joint product advertising in online videos. Manag Sci 2015;61:1921–37.
[6] Golari M, Fan N, Jin T. Multistage stochastic optimization for production-inventory planning with intermittent renewable energy. Prod Oper Manag 2017;26(3):409–25.
[7] Alghazi A, Elazouni A, Selim S. Improved genetic algorithm for finance-based scheduling. J Comput Civ Eng 2012;27(4):379–94.
[8] Elazouni AM, Metwally FG. Finance-based scheduling: tool to maximize project profit using improved genetic algorithms. J Constr Eng Manag 2005;131(4):400–12.
[9] Guillén G, Badell M, Espuna A, Puigjaner L. Simultaneous optimization of process operations and financial decisions to enhance the integrated planning/scheduling of chemical supply chains. Comput Chem Eng 2006;30(3):421–36.
[10] Elazouni AM, Metwally FG. Finance-based scheduling: tool to maximize project profit using improved genetic algorithm. J Constr Eng Manag 2005;131(4):400–12.
[11] Daniels RL, Mazzola JB, Shi D. Flow shop scheduling with partial resource flexibility. Manag Sci 2004;50(5):658–69.
[12] Alcaraz J, Maroto C. A robust genetic algorithm for resource allocation in project scheduling. Ann Oper Res 2001;102(1–4):83–109.
[13] Zhan ZH, Liu XF, Gong YJ, Zhang J, Chung HSH, Li Y. Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Comp Surv (CSUR) 2015;47(4):63.
[14] Chen PH, Weng H. A two-phase GA model for resource-constrained project scheduling. Autom Constr 2009;18(4):485–98.
[15] Niu Y, Wu X, Yang H, Zhang H. Co-design of Discrete Linear Quadratic Regulator and Resource Scheduling for Priority-Driven Networked Control Systems. In: Computational Science and Computational Intelligence (CSCI), 2014 International Conference on, vol. 1. IEEE; March, 2014. p. 15–20.
[16] Barketau MS, Cheng TCE, Kovalyov MY. Batch scheduling of deteriorating reworkables. Eur J Oper Res 2008;189(3):1317–26.
[17] Mosheiov G, Oron D. A single machine batch scheduling problem with bounded batch size. Eur J Oper Res 2008;187(3):1069–79.
[18] Liu M, Wang S, Chu C. Scheduling deteriorating jobs with past-sequence-dependent delivery times. Int J Prod Econ 2013;144(2):418–21.
[19] Do Chung B, Kim BS. A hybrid genetic algorithm with two-stage dispatching heuristic for a machine scheduling problem with step-deteriorating jobs and rate-modifying activities. Comput Ind Eng 2016;98:113–24.
[20] Gupta JND, Gupta SK. Single facility scheduling with nonlinear processing times. Comput Ind Eng 1988;14(4):387–93.
[21] Wang JB. Single machine scheduling with a time-dependent learning effect and deteriorating jobs. J Oper Res Soc 2009;60(4):583–6.
[22] Yin Y, Wu WH, Cheng TCE, Wu CC. Single-machine scheduling with time-dependent and position-dependent deteriorating jobs. Int J Comp Integ M 2015;28(7):781–90.
[23] Pei J, Pardalos PM, Liu X, Fan W, Yang S. Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. Eur J Oper Res 2015;244(1):13–25.
[24] Luo W, Ji M. Scheduling a variable maintenance and linear deteriorating jobs on a single machine. Inf Process Lett 2015;115(1):33–9.
[25] Shahvari O, Logendran R. An enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes. Comput Oper Res 2017;77:154–76.
[26] Arbib C, Marinelli F, Pezzella F. An LP-based tabu search for batch scheduling in a cutting process with finite buffers. Int J Prod Econ 2012;136(2):287–96.
[27] Vallada E, Ruiz R. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. Eur J Oper Res 2011;211(3):612–22.
[28] Castillo F, Gazmuri P. Genetic algorithms for batch sizing and production scheduling. Int J Adv Manuf Technol 2015;77(1–4):261–80.
[29] Lee WC, Wu CC, Hsu PH. A single-machine learning effect scheduling problem with release times. Omega 2010;38(1):3–11.
[30] Cheng TCE, Wu CC, Lee WC. Some scheduling problems with deteriorating jobs and learning effects. Comput Ind Eng 2008;54(4):972–82.
[31] Wang XY, Wang JJ. Single-machine due date assignment problem with deteriorating jobs and resource-dependent processing times. Int J Adv Manuf Technol 2013;67(1–4):255–60.
[32] Zhu Z, Sun L, Chu F, Liu M. Single-machine group scheduling with resource allocation and learning effect. Comput Ind Eng 2011;60(1):148–57.
[33] Wei CM, Wang JB, Ji P. Single-machine scheduling with time-and-resource-dependent processing times. Appl Math Model 2012;36(2):792–8.
[34] He H, Liu M, Wang JB. Resource constrained scheduling with general truncated job-dependent learning effect. J Comb Optim 2017;33(2):626–44.
[35] Wang JB, Wang LY, Wang D, Wang XY. Single-machine scheduling with a time-dependent deterioration. Int J Prod Econ 2009;43(7–8):805–9.
[36] Ji M, Cheng TCE. Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan. Eur J Oper Res 2010;202(1):90–8.
[37] Kuo WH, Yang DL. Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect. J Oper Res Soc 2008;59(3):416–20.
[38] Ji M, Ge J, Chen K, Cheng TCE. Single-machine due-window assignment and scheduling with resource allocation, aging effect, and a deteriorating rate–modifying activity. Comput Ind Eng 2013;66(4):952–61.
[39] Pei J, Liu X, Pardalos PM, Fan W, Yang S. Single machine serial-batching scheduling with independent setup time and deteriorating job processing times. Optim Lett 2015;9(1):91–104.
[40] Pei J, Liu X, Pardalos PM, Migdalas A, Yang S. Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine. J Global Optim 2017;67(1-2):251–62.
[41] Fan W, Pei J, Liu X, Pardalos PM, Kong M. Serial-batching group scheduling with release times and the combined effects of deterioration and truncated job-dependent learning. J Global Optim 2017. doi:10.1007/s10898-017-0536-7.
[42] Pei J, Liu X, Pardalos PM, Fan W, Yang S. Scheduling deteriorating jobs on a single serial-batching machine with multiple job types and sequence-dependent setup times. Ann Oper Res 2017;249(1-2):175–95.
[43] Pei J, Cheng B, Liu X, Pardalos PM, Kong M. Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. Ann Oper Res 2017. doi:10.1007/s10479-017-2481-8.
[44] Pei J, Liu X, Liao B, Pardalos PM, Kong M. Single-machine scheduling with learning effect and resource-dependent processing times in the serial-batching production. Appl Math Model, DOI:10.1016/j.apm.2017.07.028.
[45] Xuan H, Tang LX. Scheduling a hybrid flowshop with batch production at the last stage. Comput Oper Res 2007;34(9):2718–33.
[46] Wei C-M, Wang J-B, Ji P. Single-machine scheduling with time-and-resource-dependent processing times. Appl Math Model 2012;36(2):792–8.
[47] Wang X-Y, Wang J-J. Single-machine due date assignment problem with deteriorating jobs and resource-dependent processing times. Int J Adv Manuf Technol 2013;67(1–4):255–60.
[48] Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG. Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discret Math 1979;5:287–326.
[49] Coffman EG Jr, Garey MR, Johnson DS. An application of bin-packing to multiprocessor scheduling. SIAM J Comp 1978;7(1):1–17.
[50] Hansen P, Mladenovic N, Brimberg J, Moreno Perez J. Variable neighborhood search. In: Glover, Kochenberger, editors. Handbook of metaheuristics. London: Kluwer Academic Publishers; 2003. p. 621–757.
[51] Jarboui B, Eddaly M, Siarry P. A hybrid genetic algorithm for solving no-wait flowshop scheduling problems. Int J Adv Manuf Technol 2011;54(9):1129–43.
[52] Kovačević D, Mladenović N, Petrović B, Milošević P. DE-VNS: self-adaptive differential evolution with crossover neighborhood search for continuous global optimization. Comput Oper Res 2014;52:157–69.
[53] Amaldass N, Lucas C, Mladenovic N. A heuristic hybrid framework for vector job scheduling. Yugosl J Oper Res 2017;27:31–45.
[54] Djenic A, Radojicic N, Maric M, Mladenovic M. Parallel VNS for bus terminal location problem, Appl Soft Comput 42, 448–458.
[55] Hansen P, Mladenovic N, Todosijevic R, Hanafi S. Variable neighborhood search: basics and variants. EURO J Comput Optim 2017;5:423–54.
[56] Hanafi S, Lazic J, Mladenovic N, Wilbaut C, Crevits I. New VNS based 0-1 MIP heuristics. Yugosl J Oper Res 2015;25:343–60.
[57] Yang XS. A New Metaheuristic Bat-Inspired Algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N, editors. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol. 284. Berlin, Heidelberg: Springer; 2010.
[58] Marichelvam MK, Prabaharan T, Yang XS, Geetha M. Solving hybrid flow shop scheduling problems using bat algorithm. Int J Logist Econ Glob 2013;5(1):15–29.
[59] Osaba E, Yang XS, Diaz F, Lopez-Garcia P, Carballedo R. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Eng Appl Artif Intell 2016;48:59–71.
[60] Pei J, Liu X, Pardalos PM, Fan W, Yang S, Wang L. Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain. Int J Adv Manuf Technol 2014;70(1-4):335–48.
[61] Gandomi AH, Yang XS, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2013;22(6):1239–55.
[62] AitZai A, Benmedjdoub B, Boudhar M. Branch-and-bound and PSO algorithms for no-wait job shop scheduling. J Intell Manuf 2016;27(3):679.