INFORMATICA, 2018, Vol. 29, No. 2, 281–301
© 2018 *Vilnius University*
DOI: http://dx.doi.org/10.15388/Informatica.2018.168

Single-Machine and Parallel-Machine Parallel-Batching Scheduling Considering Deteriorating Jobs, Various Group, and Time-Dependent Setup Time

Baoyu LIAO[1,3], Jun PEI[1,2]*, Shanlin YANG[1,3], Panos M. PARDALOS[2], Shaojun LU[1,3]

[1]*School of Management, Hefei University of Technology, Hefei, China*
[2]*Center for Applied Optimization, Department of Industrial and Systems Engineering University of Florida, Gainesville, USA*
[3]*Key Laboratory of Process Optimization and Intelligent Decision-Making of Ministry of Education, Hefei, China*
e-mail: feiyijun.ufl@gmail.com

Received: May 2017; accepted: February 2018

Abstract. This paper studies a set of novel integrated scheduling problems by taking into account the combinatorial features of various groups, parallel-batching, deteriorating jobs, and time-dependent setup time simultaneously under the settings of both single-machine and parallel-machine, and the objective of the studied problems is to minimize the makespan. In order to solve the single-machine scheduling problem, we first investigate the structural properties on jobs sequencing, jobs batching, and batches sequencing for the optimal solution, and then develop a scheduling rule. Moreover, for solving the parallel-machine scheduling problem, we exploit the optimal structural properties and batching rule, and propose a novel hybrid AIS-VNS algorithm incorporating Artificial Immune System algorithm (AIS) and Variable Neighbourhood Search (VNS). Extensive computational experiments are conducted to evaluate the performance of the proposed AIS-VNS algorithm, and comparison results show that the proposed algorithm performs quite well in terms of both efficiency and solution quality.

Key words: scheduling, parallel-batching, group scheduling, deterioration, time-dependent setup time.

1. Introduction

The parallel-batching scheduling models are well-known in many practical applications of the real-life manufacturing systems. In these models, parallel-batching machines allow several jobs to be processed simultaneously in a batch, and the processing time of a batch is equal to the largest processing time of the jobs in that batch (Arroyo and Leung, 2017). The total size of a batch cannot exceed the capacity of the parallel-batching machine, and

*Corresponding author.

all jobs in each batch have the same completion time (Abedi *et al.*, 2015). Recent research on parallel-batching scheduling problems can be found in Xuan and Tang (2007), Wang and Tang (2008), Tang and Gong (2009), Tang and Liu (2009), Fan *et al.* (2012), Hazir and Kedad-Sidhoum (2014), Liu *et al.* (2014), Li and Yuan (2014), Jia and Leung (2015), Li *et al.* (2015), He *et al.* (2016), Li *et al.* (2017), Ham (2017), and Ozturk *et al.* (2017), etc.

In many practices, production scheduling problems involve deteriorating jobs, where a job to be processed later requires longer processing time (Wang and Wang, 2014). As a practical scheduling problem in a steel plant, the longer an ingot waits until the start of the preheating processing time, the more time the steel-making process will take to preheat the ingots (Li *et al.*, 2011). Recently, Yin *et al.* (2017) addressed the parallel-machine scheduling problems of deteriorating jobs with potential machine disruptions, and developed pseudo-polynomial-time solution algorithms and fully polynomial-time approximation schemes to solve two different cases. Lalla-Ruiz and Voß (2016) investigated the parallel machine scheduling problem with step deteriorating jobs, and proposed two novel mathematical models based on the Set Partitioning Problem. Tang *et al.* (2017) addressed a single-machine scheduling problems with two agents and deteriorating jobs, and proposed multiple efficient algorithms for certain special cases. More recent papers considering scheduling jobs with deteriorating jobs include Lu (2016), Yang *et al.* (2015), Yue and Wan (2016), Su and Wang (2017), Zhang *et al.* (2017), Wang and Li (2017), and Bai *et al.* (2014).

We have also investigated the serial-batching scheduling problems with deteriorating jobs in our previous research (Pei *et al.*, 2015a, 2015b, 2017a, 2017b, 2017c, Liu *et al.*, 2017). There are some key differences among the previous research and this paper: (1) we focus on parallel-batching scheduling in this paper, while we mainly studied the serial-batching scheduling in our previous papers; (2) Different from other researchers' previous research (Li *et al.*, 2011), various groups are further investigated, and the time-dependent setup time is required for processing each group and batch; (3) A new deteriorating function of job processing time is considered for the parallel-batching scheduling, and both single-machine and parallel-machine cases are studied.

To the best of our knowledge, there is no previous research considering these features simultaneously. The main contributions of this paper can be summarized as follows:

(1) We study novel integrated scheduling problems by taking into account the combinatorial features of various groups, parallel-batching, deteriorating jobs, and time-dependent setup time simultaneously under the settings of both single-machine and parallel-machine.
(2) For solving the single-machine scheduling problem, the structural properties on jobs sequencing in the same and different batches, jobs number argument, and batches sequencing are first proposed. Then, a scheduling rule is developed to solve this problem based on these structural properties.
(3) For solving the parallel-machine scheduling problem, we develop a novel hybrid AIS-VNS algorithm incorporating Artificial Immune System algorithm (AIS) and Variable Neighbourhood Search (VNS), and the optimal structural properties and

scheduling rules for the single-machine setting are integrated in this hybrid algorithm.

The remainder of this paper is organized as follows. The notations and problem description are given in Section 2. The single-machine and parallel-machine scheduling problems are studied in Sections 3 and 4, respectively. Finally, we conclude the paper in Section 5.

2. Notation and Problem Description

The notation used throughout this paper is first described in Table 1.

Given a set of N non-preemptively deteriorating jobs to be processed, all jobs are classified into n groups in advance, and each group contains a certain number of jobs, that is, $G_i = \{J_{i1}, J_{i2}, \dots, J_{N_i}\}$, $i = 1, 2, \dots, n$. In this paper, we address parallel-batching scheduling problems of a single and parallel-batching machines respectively, considering deteriorating jobs and time-dependent setup time based on various groups and minimizing the makespan as the objective. In the single-machine scheduling problem, all jobs in each group are batched and processed on a single parallel-batching machine. In the parallel-machine scheduling, all jobs in each group are first assigned into parallel machines and then processed on each machine. During the parallel-batching processing, all the jobs

Table 1
The list of the notation.

Notation	Definition
n	The number of job groups
G_i	The job set of group i , $i = 1, 2, \dots, n$
N_i	The number of jobs in G_i, $i = 1, 2, \dots, n$
N	The total number of jobs, i.e. $N = N_1 + N_2 + \cdots + N_n$
J_{ij}	Job j in G_i, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, N_i$
p_{ij}	The normal processing time of J_{ij}, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, N_i$
b	The deteriorating rate of processing jobs
p_{ij}^A	The actual processing time of J_{ij}, $j = 1, 2, \dots, N_i$, $i = 1, 2, \dots, n$
m_i	The number of batches in G_i, $i = 1, 2, \dots, n$
b_{ik}	Batch k in G_i, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
A_{ik}	The normal processing time of b_{ik}, $A_{ik} = \max\{p_{ij} : J_{ij} \in b_{ik}\}$, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
n_{ik}	The number of jobs in b_{ik}, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
θ_b	The deteriorating rate of batches' setup times
θ_g	The deteriorating rate of groups' setup times
s_b^{ik}	The setup time of b_{ik}, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
s_g^i	The setup time of G_i, $i = 1, 2, \dots, n$
c	The capacity of the batching machine
$S(b_{ik})$	The starting time of b_{ik}, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
$C(b_{ik})$	The completion time of b_{ik}, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
$P(b_{ik})$	The actual processing time of b_{ik}, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$
$C(G_i)$	The completion time of G_i, $i = 1, 2, \dots, n$
$P(G_i)$	The total actual processing time of G_i, $i = 1, 2, \dots, n$
C_{max}	The makespan

within a batch are processed simultaneously (Jia *et al.*, 2015), and the completion times of all jobs are equal to that of their belonged batch. It is also required that the maximum number of jobs in a batch cannot exceed the capacity of the parallel-batching machine.

In this paper, we further extend the application of Cheng and Ding (2000) and Cheng *et al.* (2004) in the parallel-batching scheduling problem combining various groups. If J_{ij} is scheduled at the time t, then its actual processing time is defined as Cheng and Ding (2000), Cheng *et al.* (2004)

$$p_{ij}^{A} = p_{ij} + bt, \quad j = 1, 2, \dots, N_i, \ i = 1, 2, \dots, n$$

where p_{ij} is the normal processing time of J_{ij}, $b > 0$ is the deteriorating rate of processing jobs, and t is the starting time for processing J_{ij}.

Furthermore, the normal processing time of b_{ik} is defined as $A_{ik} = \max\{p_{ij} : J_{ij} \in b_{ik}\}$, $k = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$. Then, its actual processing time can be obtained by $A_{ik} + bt$ where t is the starting time for processing b_{ik}.

Setup time is required before processing one group or batch, and the setup times of G_i and b_{ik} are defined as follows:

$$s_g^i = \theta_g t,$$
$$s_b^{ik} = \theta_b t',$$

where θ_g and θ_b are the deteriorating rates of setup time for batches and groups, and t and t' are the starting time of processing G_i and b_{ik}, respectively.

In the remaining sections of the paper, all the problems are denoted by the three-field notation schema $\alpha|\beta|\gamma$ introduced by Graham *et al.* (1979).

3. Problem $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$

In this section, we first focus on the single-machine scheduling case. The completion time of the makespan is first given, and the structural properties on jobs sequencing in the same and different batches, jobs number argument, and batches sequencing are proposed. Then, a scheduling rule is developed to solve this problem.

Lemma 1. *For the problem $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, given any schedule $\pi = (G_1, G_2, \dots, G_n)$ with all groups arriving at time $t_0 > 0$, if the starting time of G_1 $(f = 1, 2, \dots, n)$ is t_0, then the makespan is*

$$\begin{aligned} C_{\max}(\pi) = {} & t_0(1+\theta_g)^n (1+\theta_b)^{\sum_{i=1}^{n} m_i} (1+b)^{\sum_{i=1}^{n} m_i} \\ & + \sum_{i=1}^{n} (1+\theta_g)^{n-i} (1+\theta_b)^{\sum_{f=i+1}^{n} m_i} (1+b)^{\sum_{f=i+1}^{n} m_i} \\ & \times \sum_{k=1}^{m_i} (1+\theta_b)^{m_i-k} (1+b)^{m_i-k} A_{ik}. \end{aligned} \tag{1}$$

Proof. The proof of this lemma can be completed by the mathematical induction. Firstly for $n = 1$, it can be derived that

$$C(G_1) = t_0(1+\theta_g)(1+\theta_b)^{m_1}(1+b)^{m_1} + \sum_{k=1}^{m_1}(1+\theta_b)^{m_1-k}(1+b)^{m_1-k}A_1k.$$

It can be seen that Eq. (1) holds for $n = 1$. Suppose that for all $2 \leqslant d \leqslant n-1$, if Eq. (1) is satisfied, then it can be obtained that

$$\begin{aligned} C(G_d) &= t_0(1+\theta_g)^d(1+\theta_b)^{\sum_{i=1}^d m_i}(1+b)^{\sum_{i=1}^d m_i} \\ &\quad + \sum_{i=1}^{d}(1+\theta_g)^{d-i}(1+\theta_b)^{\sum_{f=i+1}^d m_i}(1+b)^{\sum_{f=i+1}^d m_i} \\ &\quad \times \sum_{k=1}^{m_i}(1+\theta_b)^{m_i-k}(1+b)^{m_i-k}A_{ik}. \end{aligned}$$

Then,

$$\begin{aligned} C(G_{d+1}) &= C(G_d)(1+\theta_g)(1+\theta_b)^{m_{d+1}}(1+b)^{m_{d+1}} \\ &\quad + \sum_{k=1}^{m_{d+1}}(1+\theta_b)^{m_{d+1}-k}(1+b)^{m_{d+1}-k}A_{(d+1)k} \\ &= t_0(1+\theta_g)^{d+1}(1+\theta_b)^{\sum_{i=1}^{d+1} m_i}(1+b)^{\sum_{i=1}^{d+1} m_i} \\ &\quad + \sum_{i=1}^{d+1}(1+\theta_g)^{d+1-i}(1+\theta_b)^{\sum_{f=i+1}^{d+1} m_i}(1+b)^{\sum_{f=i+1}^{d+1} m_i} \\ &\quad \times \sum_{k=1}^{m_i}(1+\theta_b)^{m_i-k}(1+b)^{m_i-k}A_{ik}. \end{aligned}$$

Thus, Eq. (1) still holds for $n = d+1$, and it can be also derived that Eq. (1) holds for G_n. The proof is completed. □

Lemma 2. *For the problem* $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, *all batches in the same group* G_i $(i = 1, 2, \ldots, n)$ *should be sequenced in non-decreasing order of* A_{ik} $(k = 1, 2, \ldots, m_i)$ *in the optimal schedule.*

Proof. Here we assume that π^* and π are an optimal schedule and a job schedule, respectively. The difference of these two schedules is the pairwise interchange of these two batches b_{dl} and $b_{d(l+1)}$ $(l = 1, 2, \ldots, m_i - 1,\ d = 1, 2, \ldots, n)$, that is, $\pi^* = (W_1, b_{dl}, b_{d(l+1)}, W_2)$, $\pi = (W_1, b_{d(l+1)}, b_{dl}, W_2)$, where $b_{dl}, b_{d(l+1)} \in G_d$, $m_d \geqslant 2$. W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. It is assumed that $A_{dl} > A_{d(l+1)}$.

We first give the completion time of b_v in π^*,

$$\begin{aligned}C(b_{d(l+1)}(\pi^*)) = {} & t_0(1+\theta_g)^d(1+\theta_b)^{\sum_{i=1}^{d-1} m_i+l+1}(1+b)^{\sum_{i=1}^{d-1} m_i+l+1}+(1+\theta_g)\\ & \times (1+b)^{l+1}\sum_{i=1}^{d-1}(1+\theta_g)^{d-i}(1+\theta_b)^{\sum_{f=i+1}^{d} m_i}(1+b)^{\sum_{f=i+1}^{d} m_i}\\ & \times \sum_{k=1}^{m_i}(1+\theta_b)^{m_i-k}(1+b)^{m_i-k}A_{ik}\\ & +\sum_{k=1}^{l+1}(1+\theta_b)^{l+1-k}(1+b)^{l+1-k}A_{dk}.\end{aligned}$$

Then, the completion time of b_v in π is

$$\begin{aligned}C(b_{dl}(\pi)) &= C(b_{d(l+1)}(\pi^*))\\ &= t_0(1+\theta_g)^d(1+\theta_b)^{\sum_{i=1}^{d-1} m_i+l+1}(1+b)^{\sum_{i=1}^{d-1} m_i+l+1}+(1+\theta_g)\\ &\quad \times (1+b)^{l+1}\sum_{i=1}^{d-1}(1+\theta_g)^{d-i}(1+\theta_b)^{\sum_{f=i+1}^{d} m_i}(1+b)^{\sum_{f=i+1}^{d} m_i}\\ &\quad \times \sum_{k=1}^{m_i}(1+\theta_b)^{m_i-k}(1+b)^{m_i-k}A_{ik}+\sum_{k=1}^{l+1}(1+\theta_b)^{l+1-k}(1+b)^{l+1-k}A_{dk}\\ &\quad -(1+b)A_{dl}-A_{d(l+1)}+(1+b)A_{d(l+1)}+A_{dl}.\end{aligned}$$

Consequently,

$$\begin{aligned}C\big(b_{d(l+1)}(\pi^*)\big)-C\big(b_{dl}(\pi)\big) &= (1+b)A_{dl}+A_{d(l+1)}-\big[(1+b)A_{d(l+1)}+A_{dl}\big]\\ &= b(A_{dl}-A_{d(l+1)}).\end{aligned}$$

Since $A_{dl} > A_{d(l+1)}$, it can be obtained that $C(b_{d(l+1)}(\pi^*)) > C(b_{dl}(\pi))$, which conflicts with the optimal schedule. Hence, it should be $A_{dl} \leqslant A_{d(l+1)}$.

The proof is completed. □

Based on Lemma 2, the following properties of the jobs sequencing in the different batch can be further derived.

Lemma 3. *For the problem* $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$*, all jobs in the same group* G_i $(i = 1, 2, \dots, n)$ *should be sequenced in non-decreasing order of* p_{ij} $(j = 1, 2, \dots, N_i)$ *in the optimal schedule.*

The proof is similar to that of Lemma 2, and we omit it.

Next we derive the properties on the argument of jobs number in all batches from the same group.

Lemma 4. *For the problem* $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, *it should be* $n_{il} \leqslant n_{i(l+1)}$ *for all batches from a certain group* G_i, *where* $i = 1, 2, \ldots, n$, $l = 1, 2, \ldots, m_i - 1$.

The proof can be completed based on jobs transferring operations. It is similar to that of Lemma 2, and we omit it.

Then, the following property on the argument of batches number can be further obtained.

Lemma 5. *For the problem* $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, *there should be* $\lceil \frac{N_i}{c} \rceil$ *batches and all batches are full of jobs except possibly the first batch for any group* G_i *in the optimal schedule, where* $i = 1, 2, \ldots, n$.

Then, based on Lemmas 2–5, we develop the following optimal jobs batching policy of each group for the problem $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$.

Optimal Policy 1

```
Step 1. Set $i = 1$.
Step 2. All jobs are indexed in the non-decreasing order of $p_{ij}$ in $G_i$, $j = 1, 2, \ldots, N_i$,
        and a job list is generated such that $p_{i1} \leqslant p_{i2} \leqslant \cdots \leqslant p_{iN_i}$.
Step 3. Place the first $N_i - (\lceil \frac{N_i}{c} \rceil - 1)c$ jobs in the first batch of $G_i$.
Step 4. If there are more than $c$ jobs in the job list, then place $c$ jobs in the batch and
        iterate. Then, all batches are generated in $G_i$.
Step 5. If $i < n$, then set $i = i + 1$, go to step 2. Otherwise, end.
```

According to the optimal jobs batching policy of each group, we can further obtain the following property for the optimal sequencing of each group.

Lemma 6. *For the problem* $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, *considering two consecutive groups* G_r *and* G_{r+1}, *if* $\rho(G_r) \leqslant \rho(G_{r+1})$, *where* $\rho(G_r) = \frac{\sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}}{(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}}$, $r = 1, 2, \ldots, n-1$, *then it is optimal to process* G_r *before* G_{r+1}.

Proof. Let π^* and π be an optimal schedule and a job schedule, and their difference is the pairwise interchange of these two job sets G_r and G_{r+1} $(r = 1, 2, \ldots, n-1)$, that is, $\pi^* = (W_1, G_r, G_{r+1}, W_2)$, $\pi = (W_1, G_{r+1}, G_r, W_2)$, where both G_r and G_{r+1} may include one or multiple batches, W_1 and W_2 represent two partial sequences, and W_1 or W_2 may be empty. Here we assume that $\rho(G_r) > \rho(G_{r+1})$, i.e.

$$\frac{\sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}}{(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}} > \frac{\sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}}{(1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}}},$$

and we denote the starting time of processing G_r as T.

For π^*, the completion time of G_{r+1} is

$$\begin{aligned}
&C(G_{r+1}(\pi^*)) \\
&= \left[(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}T + \sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}\right] \\
&\quad \times (1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}} + \sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}.
\end{aligned}$$

For π, the completion time of G_r is

$$\begin{aligned}
&C(G_r(\pi)) \\
&= \left[(1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}}T + \sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}\right] \\
&\quad \times (1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r} + \sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}.
\end{aligned}$$

Then,

$$\begin{aligned}
&C(G_{r+1}(\pi^*)) - C(G_r(\pi)) \\
&= \left[(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}T + \sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}\right] \\
&\quad \times (1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}} + \sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k} \\
&\quad - \left[(1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}}T \right. \\
&\quad \left. + \sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}\right] \\
&\quad \times (1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r} - \sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk} \\
&= [(1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}} - 1]\sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk} \\
&\quad - [(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r} - 1]\sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}.
\end{aligned}$$

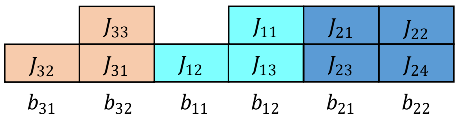


Fig. 1. The schemes of the simulation case on a single machine.

Since we have

$$\frac{\sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}}{(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}} > \frac{\sum_{k=1}^{m_{r+1}}(1+\theta_b)^{m_{r+1}-k}(1+b)^{m_{r+1}-k}A_{(r+1)k}}{(1+\theta_g)(1+\theta_b)^{m_{r+1}}(1+b)^{m_{r+1}}}.$$

Then, it can be derived that

$$C\big(G_{r+1}(\pi^*)\big) - C\big(G_r(\pi)\big).$$

It conflicts with the optimal schedule. Consequently, the proof is completed. □

Based on the above lemmas and the optimal jobs batching policy, the following Algorithm 1 can be developed to solve the problem $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$.

Algorithm 1

Step 1. Execute **Optimal Policy 1**.

Step 2. Calculate $\rho(G_r) = \frac{\sum_{k=1}^{m_r}(1+\theta_b)^{m_r-k}(1+b)^{m_r-k}A_{rk}}{(1+\theta_g)(1+\theta_b)^{m_r}(1+b)^{m_r}}$, $r = 1, 2, \ldots, n-1$.

Step 3. Sequence all groups in the non-increasing order of $\rho(G_l)$, i.e. $\rho(G_1) \leqslant \rho(G_2) \leqslant \cdots \leqslant \rho(G_n)$.

To demonstrate the algorithm 1 for the single-machine problem, a simple simulation case is given. There exist three groups, with $c = 2$, $\theta_b = 0.1$ and $\theta_g = 0.1$. The related parameters of the instance are provided in Table 2.

Table 2
The related parameters of the instance

Group (i)	1			2				3		
Job (j)	1	2	3	1	2	3	4	1	2	3
p_{ij}	0.2	0.1	0.3	0.1	0.3	0.2	0.3	0.2	0.1	0.1

The optimal schedule generated by Algorithm 1 is shown as following figure.

Theorem 1. *For the problem $1|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$, and Algorithm 1 can generate an optimal schedule in $O(N\log N)$ time.*

Proof. An optimal solution can be generated by Algorithm 1 based on Lemmas 1–6 and Corollary 1. The time complexity of step 1 is at most $O(N \log N)$, the time complexity of step 2 is $O(1)$, and for step 3 the time complexity of obtaining the optimal group sequence is $O(N \log N)$. Since we have $n \leqslant N$, the time complexity of Algorithm 1 is at most $O(N \log N)$. □

4. Problem $P|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$

Our studied problem $P|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$ can be simplified to the problem of $P_m||C_{\max}$ by relaxing that each parallel-batching machine only processes one job at a time and there is no deteriorating effect. Since Lenstra *et al.* (1977) proved that the problem of $P_m||C_{\max}$ is NP-hard, then the problem of $P|p\text{-}batch, G_i, p_{ij}^A = p_{ij} + bt|C_{\max}$ is also NP-hard. Thus, we put forward an AIS-VNS algorithm to solve it in this section and furthermore conduct some computational experiments to testify the performance of the proposed algorithm.

Since Variable Neighbourhood Search (VNS) was first proposed by Mladenović and Hansen (1997), it has been widely used in solving optimization problems. In Hansen and Mladenović (2001), they summarized the following observations: (1) A local optimal solution with respect to one neighbourhood structure is not necessary for another. (2) A local optimal solution with respect to all neighbourhood structures is the global optimal solution. (3) Different local optimal solutions with respect to different neighbourhood structures are relative to each other. The improved VNS algorithms have been put forward in previous works, for example, Wen *et al.* (2011) extended the search and improved solution quality through the combination of heuristic algorithm, VNS and genetic algorithm. Duarte *et al.* (2015) explored the adaptation of the Variable Neighbourhood Search (VNS) metaheuristic to solve multi-objective combinatorial optimization problems. Similarly, Artificial Immune System algorithm (AIS) is also an effective metaheuristic algorithm that has drawn much attention in recent years. Ying and Lin (2014) presents a novel hybrid AIS to solve the wafer sorting scheduling problem. Different mechanisms such as the clonal selection theory, the immune response along with its affinity maturation process and the immune network hypothesis can be found in Panigrahi *et al.* (2007), Castro and Zuben (2000), Komaki *et al.* (2015), and Hashemipour and Soleimani (2016). This research improves AIS through the introduction of VNS operators.

4.1. *Coding and Encoding*

In order to improve efficiency of the algorithm, we divide this problem into three stages: (1) schedule within each group to minimize the group processing time; (2) assign groups to machines; (3) sequence the groups on each machine to be processed. The proposed Optimal Policy 1 can solve the problem in the first stage. Moreover, in the third stage, the optimal processing sequence is determined by Algorithm 1. Hence, in the second stage, we use an array of group to represent a solution to our problem. Given a solution $X =$

VNS algorithm

```
Step 1. Define neighbourhood structures $N_s (s = 1, \dots, s_{\max})$.
Step 2. Get initial solution $X$.
Step 3. Execute the $s$th Local Search for $X$ to obtain a solution $X'$.
Step 4. If solution $X'$ is better than $X$, then set $x = X'$', $s = 1$ and go to step 3. Otherwise,
        set $s = s + 1$, go to step 5.
Step 5. If $s \leqslant s_{\max}$, then go to step 3; else, stop the iteration.
```

$\{2, 1, 3, 1, 3, 1\}$, these six groups are scheduled on three machines. The object suggests that jobs $\{2, 4, 6\}$, $\{1\}$, and $\{3, 5\}$ are assigned to manufacturers 1, 2, and 3, respectively, and the fitness of X is calculated by Optimal Policy 1 and Algorithm 1.

4.2. *Description of VNS Algorithm*

In order to improve the efficiency of AIS, the VNS algorithm is applied as local search strategy to strengthen the optimization ability of the algorithm, the VNS algorithm can be described as follows (Castro and Zuben, 2000):

In the VNS strategy, three types of local search operators are selected to define the neighbourhood structures N_s.

Swap operator. Given a solution X, two groups' positions x and y are selected randomly from the solution. A neighbour of X is obtained by swapping the positions x and y. The detail of swap operator in the VNS is described as follows:

The detail of swap operator in the VNS

```
Step 1. Get a solution $X$.
Step 2. Randomly generating two different numbers $x$ and $y$ from 1 to $n$.
Step 3. Select two groups' position $x$ and $y$ from the solution $X$.
Step 4. Swap the sequence of the positions $x$ and $y$ in $X$.
Step 5. Generate a neighbour of $X$.
```

Mutation operator. Given a solution X, two groups' positions x and y are selected randomly from the solution. A neighbour of X is obtained by randomly generating a number from 1 to n in the positions x and y. The detail of mutation operator in the VNS is described as follows:

The detail of mutation operator in the VNS

```
Step 1. Get a solution $X$.
Step 2. Randomly generating two different numbers $x$ and $y$ from 1 to $n$.
Step 3. Select two groups' position $x$ and $y$ from the solution $X$.
Step 4. Randomly generate a number from 1 to $n$ in the positions $x$ and $y$, respectively.
Step 5. Generate a neighbour of $X$.
```

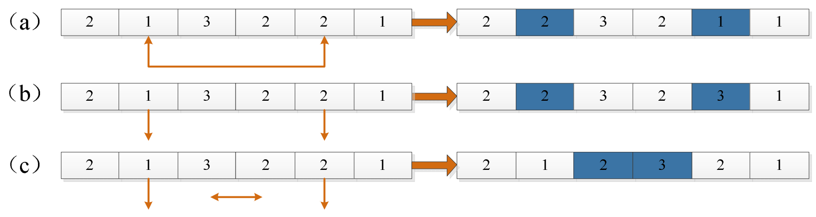


Fig. 2. The schemes of swap operator (a), mutation operator (b), and 2-opt operator (c).

2-opt operator. Given a solution X, two groups' positions x and y are selected randomly from the solution. A neighbour of X is obtained by reversal of the sequence between x and y. The detail of 2-opt operator in the VNS is described as follows:

The detail of 2-opt operator in the VNS
Step 1. Get a solution X.
Step 2. Randomly generating two different numbers x and y from 1 to n.
Step 3. Reverse the sequence between the two groups' position x and y in the solution X.
Step 4. Generate a neighbour of X.

Figure 2 shows the schemes of these three operators.

4.3. *The Algorithm Framework of AIS-VNS*

A cross operator and a variation operator are applied in AIS-VNS algorithm, $X = \{x_1, \dots, x_i, \dots, x_n\}$ and $Y = \{y_1, \dots, y_i, \dots, y_n\}$ are used to denote two solutions, and the two operators are defined as follows.

Cross operator
Step 1. Randomly select a groups' position x from the solution X.
Step 2. Replace the sequence in X before x with the sequence after x in Y.

Variation operator
Step 1. Randomly select a groups' position x from the solution X.
Step 2. Generating a number from 1 to n in position x.

We improve AIS through the introduction of VNS operators and the algorithm framework of AIS-VNS is described in Table 3. The flow chart of AIS-VNS is also given in Fig. 3.

Table 3
The Pseudocode of AIS-VNS.

Pseudocode of AIS-VNS

```
1.  Initialize parameters, including $poplong$, $PR$, $tmax$, $CM$, $pmax$ and $pmin$
2.  Set $it = 0$, $con = 0$, $gbest = a\ large\ enough\ positive\ constant$, randomly generate a population
    $pop = \{X_1, \dots, X_l, \dots, X_{poplong}\}$ with $poplong$ antibodies, and each solution includes $n$ elements,
    $X_l = \{x_{l1}, \dots, x_{li}, \dots, x_{ln}\}$
3.  While ($it \leqslant tmax$)
4.  Calculate fitness for each antibody $popfit = \{fit_1, \dots, fit_l, \dots, fit_{poplong}\}$
5.  If $gbest > \min_{1 \leqslant l \leqslant poplong} fit_l$, then
6.      Set $gbest = \min_{1 \leqslant l \leqslant poplong} fit_l$
7.  End if
8.  For population pop, $l = 1$ to $poplong$
9.      If $pmin < \frac{fit_l}{gbest} < pmax$, then
10.         Set $con = con + 1$
11.     End if
12. End for
13. If $con \leqslant CM * poplong$ then
14.     For population pop, $l = 1$ to $poplong$
15.         Set $pro = \frac{1/fit_l}{\sum_{k=1}^{prolong} 1/fit_k}$
16.     End for
17. else
18.     For population pop, $l = 1$ to $poplong$
19.         Set $pro = \frac{fit_l}{\sum_{k=1}^{prolong} fit_k}$
20.     End for
21. End if
22. For population pop, $l = 1$ to $poplong$
23.     Generate a random number $rand()$ in [0, 1]
24.     If $rand() \leqslant PR$ then
25.         Select $X_k$ from pop with probability $pro_k$, and perform variation operator for $X_k$ to get $Y_l$
26.     Else
27.         Select $X_e$ and $X_k$ from pop with probability $pro_e$ and $pro_k$, and perform cross operator for
    them to get $Y_l$
28.     End if
29.     Perform VNS operators to update $Y_l$
30.     End for
31.     For population pop, $l = 1$ to $poplong$
32.         Set $X_l = Y_l$
33.     End for
34. End while
35. Output $gbest$
```

4.4. *Computational Experiments and Comparison*

In this section, we present computational experiments to evaluate the performance of our proposed algorithm AIS-VNS, compared with AIS (Castro and Timmis, 2002), VNS (Lei, 2015), and PSO (Ercan, 2008). The test problems were randomly generated based on the real production in Table 4. Based on the number of machines and groups, 16 instances are generated in our computational experiments.

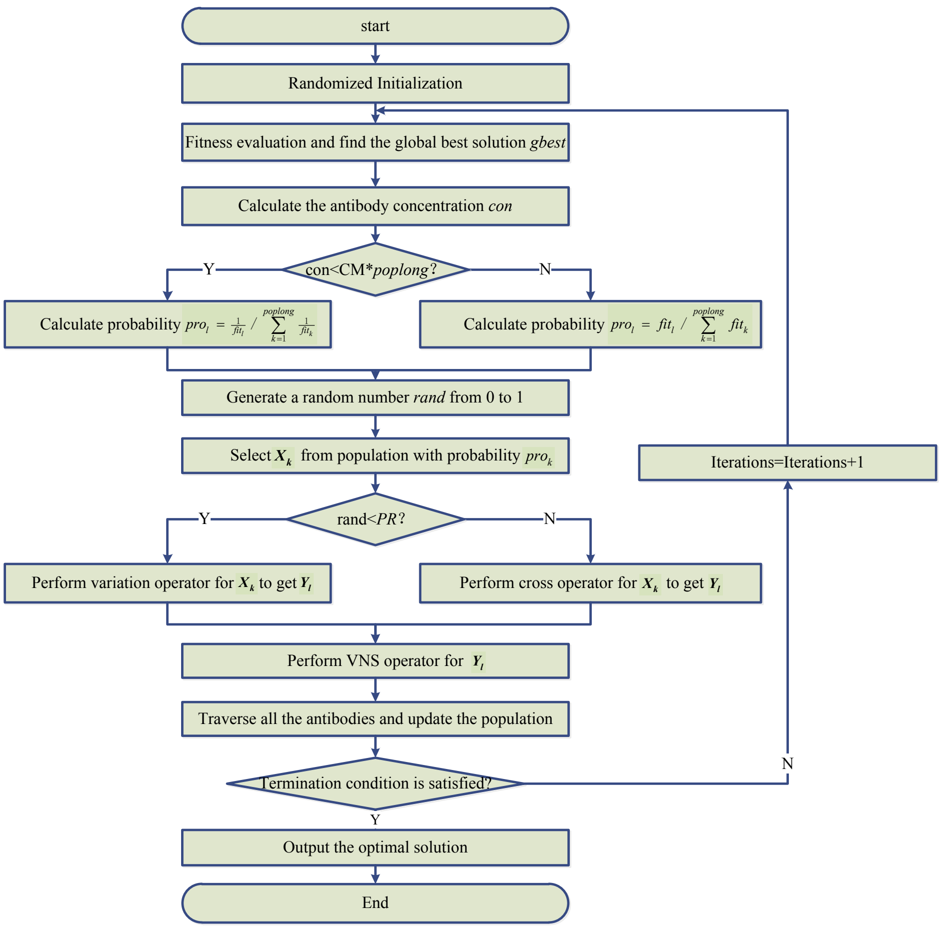

start
Randomized Initialization
Fitness evaluation and find the global best solution *gbest*
Calculate the antibody concentration *con*
con<CM**poplong*?
Y
N
Calculate probability $pro_l = \frac{1}{fit_l} / \sum_{k=1}^{poplong} \frac{1}{fit_k}$
Calculate probability $pro_l = fit_l / \sum_{k=1}^{poplong} fit_k$
Generate a random number *rand* from 0 to 1
Select X_k from population with probability pro_k
Iterations=Iterations+1
rand<*PR*?
Y
N
Perform variation operator for X_k to get Y_l
Perform cross operator for X_k to get Y_l
Perform VNS operator for Y_l
Traverse all the antibodies and update the population
Termination condition is satisfied?
N
Y
Output the optimal solution
End

Fig. 3. The flow chart of AIS-VNS.

Table 4
Parameters setting.

Notation	Definition	Value
n	The number of groups	20, 50, 100, 150
M	The number of machines	3, 5, 7, 9
θ_b	The deteriorating rate of batches' setup times	0.01
θ_g	The deteriorating rate of groups' setup times	0.01
c	The capacity of the batching machine	3
N_i	The number of jobs in $G_i, i = 1, 2, \ldots, n$	$U[1, 6]$
p_{ij}	The normal processing time of $J_{ij}, i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, N_i$	$U[0.1, 0.2]$
CM	The concentration limit in AIS	0.9
PR	The variation probability in AIS	0.9

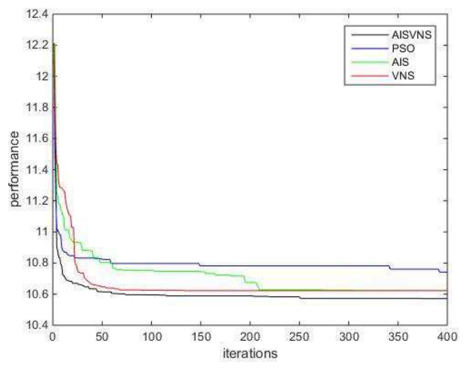
Table 5
The average objective value (Avg.Obj) and the minimum objective value (Max.Obj) for each algorithm.

n	M	AIS-VNS		AIS		VNS		PSO	
		Ave.Obj	Max.Obj	Ave.Obj	Max.Obj	Ave.Obj	Max.Obj	Ave.Obj	Max.Obj
20	3	10.571	12.399	10.623	12.518	10.622	12.663	10.742	12.628
20	5	4.683	5.198	4.751	5.248	4.970	5.691	4.805	5.270
20	7	3.289	3.547	3.425	4.065	3.666	4.745	3.441	4.076
20	9	2.570	2.814	2.667	2.880	2.846	3.318	2.746	2.876
50	3	217.085	296.108	220.559	300.668	217.848	301.268	226.132	330.299
50	5	32.665	40.856	33.121	40.914	33.461	48.288	35.945	47.115
50	7	12.561	14.954	12.976	14.931	12.618	15.653	14.409	17.915
50	9	8.137	8.858	8.452	9.115	8.622	10.016	9.529	11.228
100	3	3.490e+4	4.777e+4	3.577e+4	4.815e+4	3.513e+4	4.791e+4	3.774e+4	5.025e+4
100	5	624.447	838.807	639.009	838.911	657.050	1020.680	782.191	1020.226
100	7	109.115	135.965	113.689	140.236	116.997	139.878	133.772	177.172
100	9	43.686	46.131	45.642	51.037	46.434	54.854	61.520	73.883
150	3	4.909e+6	7.376e+6	5.005e+6	7.701e+6	4.918e+6	7.445e+6	5.486e+6	8.058e+6
150	5	1.391e+4	1.889e+4	1..432e+4	1.967e+4	1.419e+4	1.905e+4	2.001e+4	3.147e+4
150	7	899.797	1075.621	959.789	1188.779	1063.756	1475.763	1303.263	2050.683
150	9	244.268	276.081	260.709	302.718	247.808	295.556	321.250	500.599

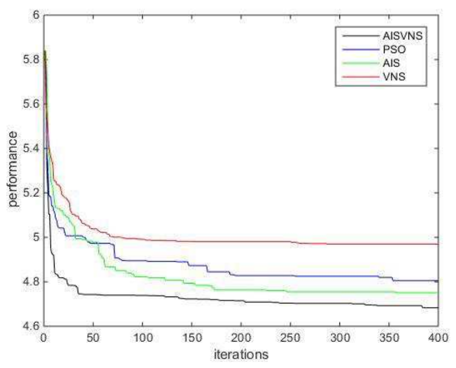
Table 5 lists the results of the average objective value (Avg.Obj) and the maximum objective value (Max.Obj) for the problem.

In this experiment, the average of the current optimal solutions for 1 to 400 iterations is used to plot the convergence curves. The convergence behaviours of each algorithm for each instance are shown in Fig. 4. (A, B) is used to denote various instances. For example, the instance with 20 jobs and 3 machines is represented by (20, 3).

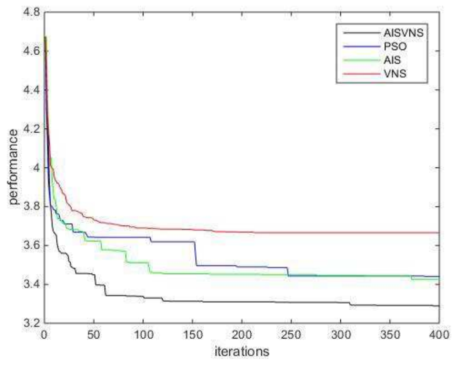
All the algorithms were implemented in C++ and run on a Lenovo computer running Windows 10 with a dual-core CPU Intel i3-3240@3.40 GHz and 4 GB RAM. As can be observed from Fig. 4, AIS-VNS outperforms all the other three methods across the 16 instances in solution quality and convergence. With respect to solution quality, it is worth mentioning that proposed algorithm obtains the best solution within 400 iterations for all the instances. Moreover, AIS-VNS also has better performance than the other algorithms in the form of convergence speed. Since proposed algorithm can converge to a reasonable solution before the 50 iterations in many instances, such as instance 1, 7, 10, 11, 12, 13, 14, 15. It is important to remark that all the available results for AIS-VNS are obtained in reasonable time. For example, per single running time is between 6 and 10 seconds when the number of jobs is 100. Thus, we can infer that the running time of AIS-VNS does not exceed 1 second in a single running. Based on the above experimental results and analysis, we can infer that the proposed AIS-VNS is stable and robust in terms of solution quality and convergence speed in solving the presented problems.



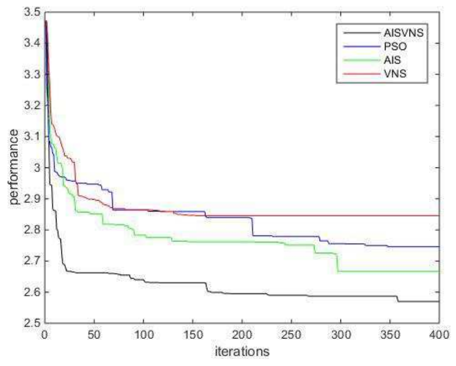
(1) Convergence curves for (20, 3)



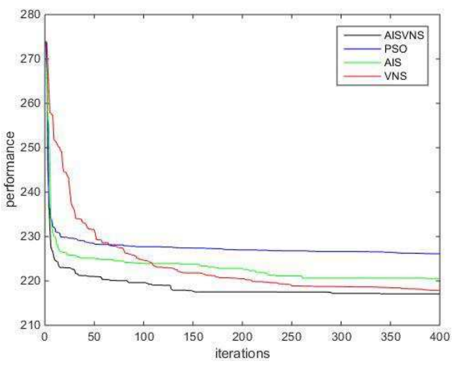
(2) Convergence curves for (20, 5)



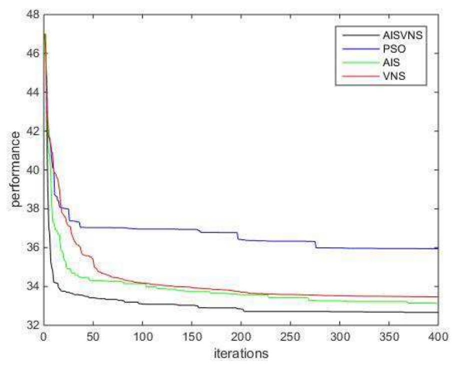
(3) Convergence curves for (20, 7)



(4) Convergence curves for (20, 9)



(5) Convergence curves for (50, 3)

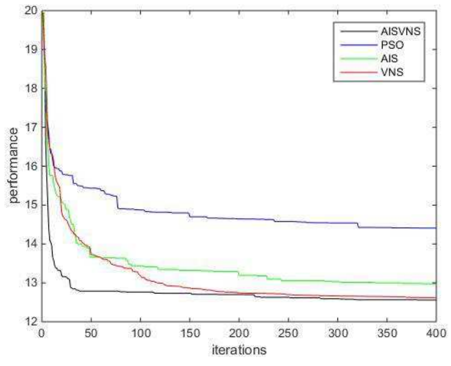


(6) Convergence curves for (50, 5)

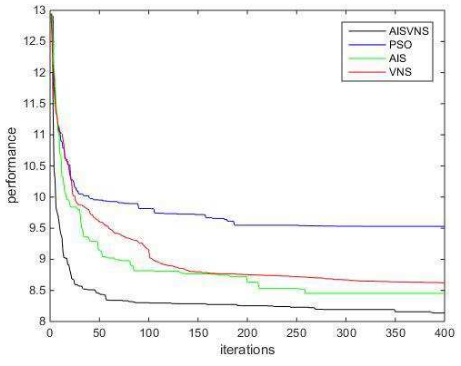
Fig. 4. Convergence curves for each instance.

5. Conclusions

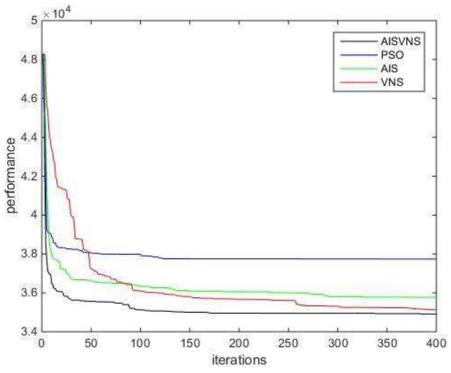
In this paper we investigate single and parallel-batching machine scheduling problems to minimize the makespan, where the combinatorial features of various groups, parallel-batching, deteriorating jobs, and time-dependent setup time are considered simultane-



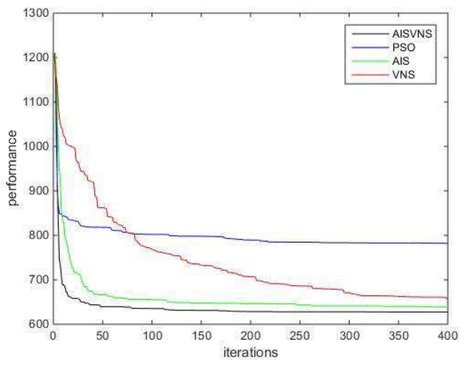
(7) Convergence curves for (50, 7)



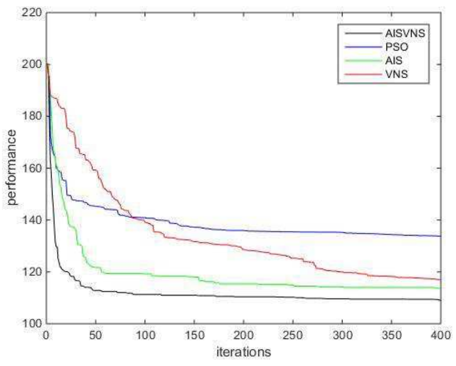
(8) Convergence curves for (50, 9)



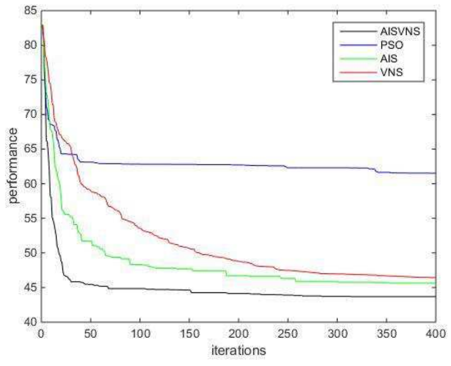
(9) Convergence curves for (100, 3)



(10) Convergence curves for (100, 5)



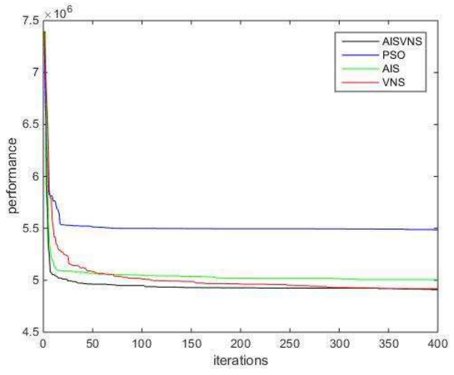
(11) Convergence curves for (100, 7)



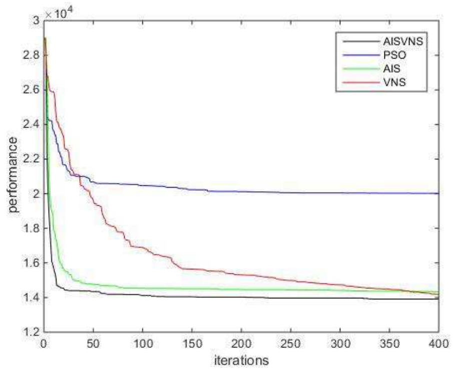
(12) Convergence curves for (100, 9)

Fig. 5. (*Continued*).

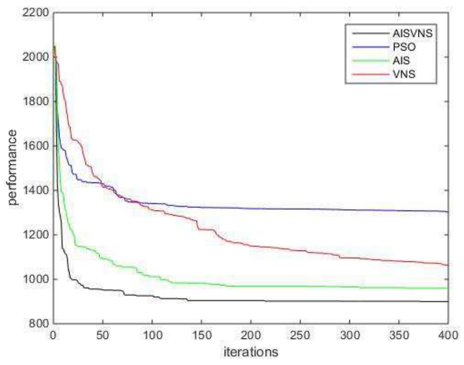
ously. For the single-machine scheduling problem, we propose the optimal structural properties and a scheduling rule to solve it. Moreover, a hybrid algorithm incorporating AIS and VNS algorithms is developed to solve the parallel-machine scheduling problem. The results of extensive computational experiments show both efficiency and solution quality of the proposed algorithm, compared with the algorithms of AIS, VNS, and PSO.



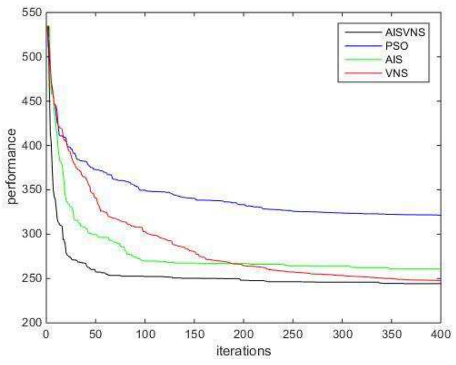
(13) Convergence curves for (150, 3)



(14) Convergence curves for (150, 5)



(15) Convergence curves for (150, 7)



(16) Convergence curves for (150, 9)

Fig. 6. (*Continued*).

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Nos. 71690230, 71601065, 71501058, 71690235), 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), and 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).

References

Abedi, M., Seidgar, H., Fazlollahtabar, H., Bijani, R. (2015). Bi-objective optimisation for scheduling the identical parallel batch-processing machines with arbitrary job sizes, unequal job release times and capacity limits. *International International Journal of Production Research*, 53(6): 1680–1711.

Arroyo, J.E.C., Leung, J.Y.-T. (2017). An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times. *Computers and Industrial Engineering*, 105, 84–100.

Bai, J., Li, Z.-R., Wang, J.-J., Huang, X. (2014). Single machine common flow allowance scheduling with deteriorating jobs and a rate-modifying activity. *Applied Mathematical Modelling*, 38(23), 5431–5438.

Castro, L.R.D., Timmis, J.L. (2002). *Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag, New York, Inc.

Castro, L.N.D., Zuben, F.J.V. (2000). *Artificial Immune Systems: Part II – A Survey of Applications*, Technical Report.

Cheng, T.C.E., Ding, Q. (2000). Single machine serial-batching scheduling with independent setup time and deteriorating job processing times. *Acta Informatica*, 36, 673–692.

Cheng, T.C.E., Ding, Q., Lin, B.M.T. (2004). A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, 152(1), 1–13.

Duarte, A., Pantrigo, J.J., Pardo, E.G., Mladenovic, N. (2015). Multi-objective variable neighborhood search: an application to combinatorial optimization problems. *Journal of Global Optimization*, 63(3), 515–536.

Ercan, M.F. (2008). A performance comparison of PSO and GA in scheduling hybrid flow-shops with multiprocessor tasks. In: *ACM Symposium on Applied Computing*. DBLP, pp. 1767–1771.

Fan, B., Yuan, J., Li, S. (2012). Bi-criteria scheduling on a single parallel-batch machine. *Applied Mathematical Modelling*, 36(3), 1338–1346.

Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. (1979). Optimization and approximation indeterministic sequencing and scheduling: a survey. *Ann. Discret. Math.* 5, 287–326.

Ham, A. (2017). Flexible job shop scheduling problem for parallel batch processing machine with compatible job families. *Applied Mathematical Modelling*, 45. doi:10.1016/j.apm.2016.12.034.

Hansen, P., Mladenović, N. (2001). Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130(3), 449–467.

Hashemipour, M.S., Soleimani, S.A. (2016). Artificial immune system based on adaptive clonal selection for feature selection and parameters optimisation of support vector machines. *Connection Science*, 1, 1–16.

Hazir, Ö., Kedad-Sidhoum, S. (2014). Batch sizing and just-in-time scheduling with common due date. *Annals of Operations Research*, 213(1), 187–202.

He, C., Leung, J.Y.T., Lee, K., Pinedo, M.L. (2016). Scheduling a single machine with parallel batching to minimize makespan and total rejection cost. *Discrete Applied Mathematics*, 2016, 204, 150–163.

Jia, Z.-H., Leung, J.Y.-T. (2015). A meta-heuristic to minimize makespan for parallel batch Machines with arbitrary job sizes. *European Journal of Operational Research*, 240, 649–665.

Jia, Z.-H., Li, K., Leung, J.Y.-T. (2015). Single machine scheduling with deadlines and increasing rates of processing times. *International Journal of Production Economics*, 169, 1–10.

Komaki, G.M., Mobin, S., Teymourian, E., Sheikh, S. (2015). A general variable neighborhood search algorithm to minimize makespan of the distributed permutation flowshop scheduling problem. 9(8), 2618–2625. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 9(8), 2618–2625.

Lalla-Ruiz, E., Voß, (2016). Modeling the parallel machine scheduling Problem with step deteriorating jobs. *European Journal of Operational Research*, 255(1), 21–33.

Lei, D. (2015). Variable neighborhood search for two-agent flow shop scheduling problem. *Computers & Industrial Engineering*, 80(C), 125–131.

Lenstra, J.K., Rinnooy, A.H.G., Brucker, P. (1977). Complexity of Machine Scheduling Problems. *Journal of Scheduling*, 1, 343–362.

Li, W., Yuan, J. (2014). Improved online algorithms for the batch scheduling of equal-length jobs with incompatible families to maximize the weighted number of early jobs. *Optimization Letters*, 8(5), 1691–1706.

Li, S., Ng, C.T., Cheng, T.C.E., Yuan, J. (2007). Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. *European Journal of Operational Research*, 210, 482–488.

Li, X., Ishii, H., Chen, M. (2015). Single machine parallel-batching scheduling problem with fuzzy due-date and fuzzy precedence relation. *International Journal of Production Research*, 53(9), 2707–2717.

Li, W., Li, S., Feng, Q. (2017). Online batch scheduling with kind release times and incompatible families to minimize makespan. *Optimization Letters*, 8(5). doi:10.1007/s11590-017-1113-1.

Liu, L.L., Ng, C.T., Cheng, T.C.E. (2014). Scheduling jobs with release dates on parallel batch processing machines to minimize the makespan. *Optimization Letters*, 8(1), 307–318.

Liu, X., Lu, S., Pei, J., Pandalos, P.M. (2017). A hybrid VNS-HS algorithm for a supply chain scheduling problem with deteriorating jobs. *International Journal of Production Research*, 1–18.

Lu, Y.-Y. (2016). Research on no-idle permutation flowshop scheduling with time-dependent learning effect and deteriorating jobs. *Applied Mathematical Modelling*, 40(4), 3447–3450.

Mladenović, N., Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097–1100.

Ozturk, O., Begen, M.A., Zaric, G.S. (2017). A branch and bound algorithm for scheduling unit size jobs on parallel batching machines to minimize makespan. *International Journal of Production Research*, 55(6), 1815–1831.

Panigrahi, B.K., Yadav, S.R., Agrawal, S., Tiwari, M.K. (2007). A clonal algorithm to solve economic load dispatch. *Electric Power Systems Research*, 77(10), 1381–1389.

Pei, J., Pardalos, P.M., Liu, X., Fan, W., Yang, S. (2015a). Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. *European Journal of Operational Research*, 244(1), 13–25.

Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S. (2015b). Single machine serial-batching scheduling with independent setup time and deteriorating job processing times. *Optimization Letters*, 9(1), 91–104.

Pei, J., Liu, X., Pardalos, P.M., Fan, W., Yang, S. (2017a). Scheduling deteriorating jobs on a single serial-batching machine with multiple job types and sequence-dependent setup times. *Annals of Operations Research*, 249, 175–195.

Pei, J., Liu, X., Pardalos, P.M., Migdalas, A., Yang, S. (2017b). Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine. *Journal of Global Optimization*, 67(1), 251–262.

Pei, J., Liu, X., Fan, W., Pardalos, P.M., Lu, S. (2017c). A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers. *Omega*. doi:10.1016/j.omega.2017c.12.003.

Su, L.-H., Wang, H.-M. (2017). Minimizing total absolute deviation of job completion times on a single machine with cleaning activities. *Computers and Industrial Engineering*, 103, 242–249.

Tang, L., Gong, H. (2009). The coordination of transportation and batching scheduling. Applied Mathematical Modelling, 33(10), 3854–3862.

Tang, L., Liu, P. (2009). Two-machine flowshop scheduling problems involving a batching machine with transportation or deterioration consideration. *Applied Mathematical Modelling*, 33(2), 1187–1199.

Tang, L., Zhao, X., Liu, J., Leung, J.Y.T. (2017). Competitive two-agent scheduling with deteriorating jobs on a single parallel-batching machine. *European Journal of Operational Research*, 263(2), 401–411.

Wang, X.-Y., Wang, J.-J. (2014). Scheduling deteriorating jobs with a learning effect on unrelated parallel machines. *Applied Mathematical Modelling*, 38(21–22), 5231–5238.

Wang, X., Tang, L. (2008). Integration of batching and scheduling for hot rolling production in the steel industry. *International Journal of Advanced Manufacturing Technology*, 36(5), 431–441.

Wang, J.B., Li, L. (2017). Machine scheduling with deteriorating jobs and modifying maintenance activities. *The Computer Journal*, 1–7.

Wen, Y. Xu, H., Yang, J. (2011). A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system. *Information Sciences*, 181(3), 567–581.

Xuan, H., Tang, L. (2007). Scheduling a hybrid flowshop with batch production at the last stage. *Computers and Operations Research*, 34(9), 2718–2733.

Yang, S.-W., Wan, L., Yin, N. (2015). Research on single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs. *Applied Mathematical Modelling*, 39(15), 4593–4598.

Yin, Y., Wang, Y., Cheng, T.C.E., Liu, W., Li, J. (2017). Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. *Omega*, 69, 17–28.

Ying, K.C., Lin, S.W. (2014). Efficient wafer sorting scheduling using a hybrid artificial immune system. *Journal of the Operational Research Society*, 65(2), 169–179.

Yue, Q., Wan, G. (2016). Single machine SLK/DIF due window assignment problem with job-dependent linear deterioration effects. *Journal of the Operational Research Society*, 67(6), 872–883.

Zhang, X., Wu, W.H., Lin, W.C., Wu, C.C. (2017). Machine scheduling problems under deteriorating effects and deteriorating rate-modifying activities. *Journal of the Operational Research Society*, 1–10.

B. Liao is currently working on his PhD degree at the School of Management, Hefei University of Technology. He obtained his master's degrees from Hefei University of Technology in 2015. His research interests include supply chain scheduling and metaheuristics.

J. Pei is currently an associate professor at the School of Management, Hefei University of Technology, Hefei, China. He is also the associate editor for *Journal of Global Optimization* (2018.1-Present), *Optimization Letters* (2016.8-Present), *Energy Systems* (2015.12-Present), *Computational Social Networks* (2016.7-Present) and a guest editor for *Journal of Combinatorial Optimization*. His research interests include supply chain scheduling, artificial intelligence, and information systems.

S. Yang is a professor at School of Management, Hefei University of Technology, Hefei, China. He is a member of Chinese Academy of Engineering. He servers as the director of National & Local Joint Engineering Research Center for Intelligent Decision and Information Systems and the director of Key Laboratory of Process Optimization and Intelligent Decision-making. His research interests include decision science and technology, optimization theory and method, management information system.

P.M. Pardalos serves as a distinguished professor of industrial and systems engineering at the University of Florida, Gainesville, FL, USA. He is also the director of the Center for Applied Optimization. Dr. Pardalos is a world leading expert in global and combinatorial optimization. His recent research interests include network design problems, optimization in telecommunications, e-commerce, data mining, biomedical applications, and massive computing.

S. Lu is currently working on his PhD degree at the School of Management, Hefei University of Technology. He obtained his BS degrees from Hefei University of Technology in 2015. His research interests include supply chain scheduling and metaheuristics.