IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 April 2024, accepted 29 April 2024, date of publication 3 May 2024, date of current version 10 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3396455

RESEARCH ARTICLE

A Job Sequence Optimization Approach for Parallel Machine Scheduling Problem in Printing Manufacturing Systems

HUAILIN LI[1,2], YINGYING ZHENG[1], BANGYONG SUN[1,2], AND BIN DU[1,2]
[1]Faculty of Printing, Packaging Engineering and Digital Media Technology, Xi'an University of Technology, Xi'an 710048, China
[2]Shaanxi Provincial Key Laboratory of Printing and Packaging Engineering, Xi'an 710048, China

Corresponding author: Huailin Li (lihuailin@xaut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62076199, and in part by the Key Research and Development Program of Shaanxi Province under Grant 2023-YBGY-087.

ABSTRACT This paper investigated a parallel machine scheduling problem in the printing manufacturing systems due to operational requirements on the color-batching of the printed matter and sequential requirements on the sequence adherence of a printing press. Resequencing printing jobs as color-oriented batches reduced the costs of color changes and operational costs for printing shop. Also, post press workshop required printing shops to print jobs with minimal makespan so that high sequence adherence with its demand is assured. Based on real-world applications, we investigated two contradictory objectives-color change costs and minimal makespan-in a parallel high-fidelity printing press scheduling environment. A job sequence optimization approach is proposed. Moving interpolation algorithm and color sequence mapping algorithm are designed to reduce frequency of replacing ink. Base on iterative greedy algorithm, color sequence job groups with the same or similar color sequence are scheduled to the printing presses. Three experiments are conducted and showed that the proposed approach has good feasibility and effectiveness, and convergence in solving minimum completion time. Therefore, the proposed approach can be used in printing shops to improve scheduling efficiency and reduce production costs.

INDEX TERMS Printing manufacturing systems, parallel machine scheduling problem, minimal makespan, color sequence comparison, iterated greedy algorithm.

I. INTRODUCTION

In recent years, market demand has changed significantly in both the printing industry [1] and the food packaging industry [2], [3]. From the original single type and large batch to multi-variety and small batch, especially the customized and high-fidelity color printing matters have attracted the attention of customers [4]. The high-fidelity (Hi-Fi) color printing matters require printing presses to have a wider color gamut range. The traditional 4-color printing method obviously cannot meet this condition. At present, the printing industry mostly uses 8-color Hi-Fi printing presses to solve this problem.

The same to traditional 4-color printing process, different 8-color Hi-Fi printing matters usually have different ink overprint sequences, which means that printing different jobs changes inks and requires a washing process to remove the previous color ink from the inner surface of the color group. This operational process is the most time-consuming stage in the printing process [5]. However, compared to the traditional 4-color printing process, the 8-color printing process has more color sequence changes (8!>4!). Therefore, the job scheduling of Hi-Fi color printing workshops is more complex. For example, a certain printing and packaging enterprise has up to 2800 types of printed matters, with a minimum production batch of 1000.

At present, most of the research focuses on the scheduling of the traditional 4-color printing workshop, while the

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Zunino.

research on the scheduling of the 8-color printing workshop is less. To solve this problem, in this paper, we research a job sequence optimization approach to minimise the makespan for the parallel color printing press scheduling problem.

Parallel machine scheduling problem (PMSP) has mathematical models, and heuristics to minimise the makespan of PMSP, such as genetic algorithm [6], variable neighborhood search algorithm [7], iterated greedy algorithms [8], [9], artificial bee colony algorithm [10], and tabu-search algorithm [11]. Adan proposed a hybrid genetic algorithm equipped with a minimal number of parameters and operators, which was enhanced with an effective local search operator, specifically targeted to solve large instances [12]. Due to the high computational complexity of the proposed model, Zandi et al. developed a heuristic algorithm to obtain the exact Pareto frontier of these two objectives with a polynomial complexity [13]. Wu and Che considered an energy-efficient bi-objective unrelated PMSP to minimise both makespan and total energy consumption [14]. Based on an attention mechanism and disjunctive graph embedding, Chen et al. proposed a novel deep reinforcement learning framework for solving the classical parallel scheduling problem, and used a sequence-to-sequence pattern to model it in the framework [15].

Because of its simple structure, few parameters and good solution effect, the iterated greedy algorithm has been widely applied to solve complex scheduling problems. Vallada et al. proposed a scatter search algorithm with an enriched iterated greedy algorithm for the unrelated PMSP, and obtained superior results, outperforming the best-known solutions [16]. Based on the iterated greedy algorithm, Tavares-Neto and Nagano designed two new algorithms, one constructive heuristic and an improvement heuristic for solving the problem of minimizing the total system makespan of an integrated production distribution system [17]. Fernandez-Viagas, Valente, and Framinan presented eight variations of an iterated greedy algorithm to minimise the total tardiness of distributed permutation flowshop scheduling problem [18]. As an easy and high-performance heuristic, the simple framework of the iterated greedy algorithm makes it easy to be implemented by practitioners, and its high performance implies its great potential to solve industrial scheduling problems [19], [20].

In the printing factory, the number of machines and workers is limited. In the face of mass processing operations, if all processing tasks can be completed in the shortest time, it means that more worker hours can be saved. Achieving the fastest delivery speed not only improves customer satisfaction, but also saves the overall cost of the factory. Thus, the parallel color press scheduling problem (PCPSP) is of great significance in the real world.

The remainder of the paper is arranged as follows. Section II gives the literature review. Section III discusses the setting principles of printing color sequence. PCPSP is described and formulated in the Section IV. Section V presents an iterated greedy algorithm combined with the moving interpolation algorithm or the color sequence mapping algorithm to solve PCPSP. In Section VI, we design and analyze the experiments. Finally, the conclusions and suggestions for the future work are discussed in Section VII.

II. RELATED WORKS

Many researchers have performed extensive research on PCPSP and have achieved good results. In this section we provide a review of the related applications.

Burger et al. solved PCPSP both exactly and heuristically for small, randomly generated test problem instances and studied the trade-off between the time efficiency and solution quality of the two approaches [21]. Schuurman and Vuuren considered PCPSP approximately by using a simple heuristic and three well-known metaheuristics (improving local search, tabu search, and simulated annealing) [22]. Iori et.al developed a greedy randomised adaptive search procedure equipped with several local search procedures for assigning print jobs to a heterogeneous set of flexographic printer machines and finding a processing sequence for the jobs assigned to each machine [5].

Lunardi et al. tackled a challenging problem in online printing shop scheduling. They presented mixed-integer linear programming and constraint programming models for minimizing the makespan [1]. Lunardi et al. also proposed and evaluated a local search strategy and meta-heuristic for a flexible job shop scheduling problem with sequencing flexibility, which was proven to have competitive performance [23]. To minimise the total printing cost composed of paper and plate costs, Mostajabdaveh et al. developed a nonlinear integer programming model to obtain exact solutions for small-sized instances and an efficient genetic algorithm to solve real-sized problems [24]. Tuyttens and Vandaele proposed a new greedy random adaptive search procedure to solve the label and the cover printing problem [25].

Besides the printing industry, the PCPSP is rich in application possibilities, such as pharmaceutical packaging facility [26], printed circuit boards (PCBs) [27], automotive paint shops [22], [27], [28], [29], the printing and dyeing industry [30], [31], personalized printing design for artworks and so on. The reason is that the color batching problem has a great impact on the scheduling in these fields. The purpose of resequencing is to obtain color-oriented batches of cars to reduce setup costs incurred in paint shops [32]. Li et.al investigated a PMSP with different color families, sequence dependent setup times, and machine eligibility restriction for the dyeing overdue problem in a lace textile factory [33]. These researchers have agreed that color-batching problem is the most time and energy-consuming process in PMSP.

In the literatures [1], [22], and [23] researchers generated small instances of print job sets for the PCPSP and assumed that each job required at least 2 but no more than 4 colors from the universal color set. Therefore, a data set of 8 colors job scheduling is established according to the principles of Hi-Fi printing color sequence in this paper.

III. PRINCIPLES OF PRINTING COLOR SEQUENCE

In the printing industry, the three primitive color inks are generally Yellow (Y), Cyan (C) and Magenta (M). Due to insufficient saturation of gray when equal amounts of tricolor inks are overprinted, Black (BK) ink is often added to compensate. In addition, adding black ink can also increase the density contrast and brightness range of the image, and reduce the amount of tricolor ink.

When printing trademarks, package, promotional materials of printed matters (such as cosmetics, drugs, daily necessities, etc.), a lot of spot color inks (such as Red, Blue, Gold, Silver, etc.) are often used to improve the design and quality of packaging and decorating products. The spot color inks can increase the gamut range of the equipment and ensure the color consistency. However, the use of spot color requires frequent ink change and printing machine cleaning. Therefore, the printing industry often uses Orange (O), Green (G), Violet (V), and White (W, or Varnish) to simulate the spot color. It not only saves the economic cost, but also shows better flexibility.

In printing shops, the color sequences directly affect the quality of printed matters. Therefore, in order to reproduce the original accurately, the following principles of Hi-Fi printing color sequence must be followed:

(1) The drying performance of the ink
Set the color sequence from slow-drying ink to fast-drying ink to increase the drying time of slow-drying ink conjunctiva.

(2) The transparency and hiding power of ink
To show the correct color and achieve good color mixing effect, the color sequence should be arranged from poor transparency to strong transparency.

(3) The lightness of tricolor ink
The higher the reflectivity, the higher the brightness of the ink. The color sequence is arranged from lower lightness to higher lightness.

(4) The ink absorption performance of printing stock
For the printing stock with roughness and poor tightness, the color sequence is set from good to poor color fluidity.

(5) The size of dot area
Generally, the color sequence is set should be arranged from small dot area to large dot area.

(6) The characteristics of the original
For printed matters with warm colors, the cyan ink should be printed first, followed by red ink and yellow ink. For the printed matters with cold colors, the printing sequence is reversed.

In conclusion, it is very important to reasonably arrange the printing color sequence to improve the quality of printed matters.

IV. PROBLEM DESCRIPTION AND ASSUMPTIONS

A. PROBLEM STATEMENT

The parallel color printing press scheduling problem can be described as having n independent printing jobs $(J_1, J_2, \ldots, J_n)$, which need to be printed on m sets of the same 8-color printing presses $(M_1, M_2, \ldots, M_m)$, and the printing time of n jobs is different, as shown in Figure 1. Now it is required to design a job scheduling scheme so that the n jobs can be printed in the shortest time. The essence of multi-machine scheduling problem is to maximize the overall benefit and do the most things in the shortest time. In this paper, we mainly focus on the scheduling problem of multi variety and small batch printing jobs (1000-2000), which can be seen in Table 4. The PCPSP can be solved with the help of a numerical illustration and formulates the mathematical model along with the following assumptions:

- All jobs are completed on the printing presses.
- All printing presses are available at the beginning of the planning horizon.
- Each job can be printed on any printing press, but it can only be printed on one printing press simultaneously.
- All jobs cannot be interrupted once being printed.

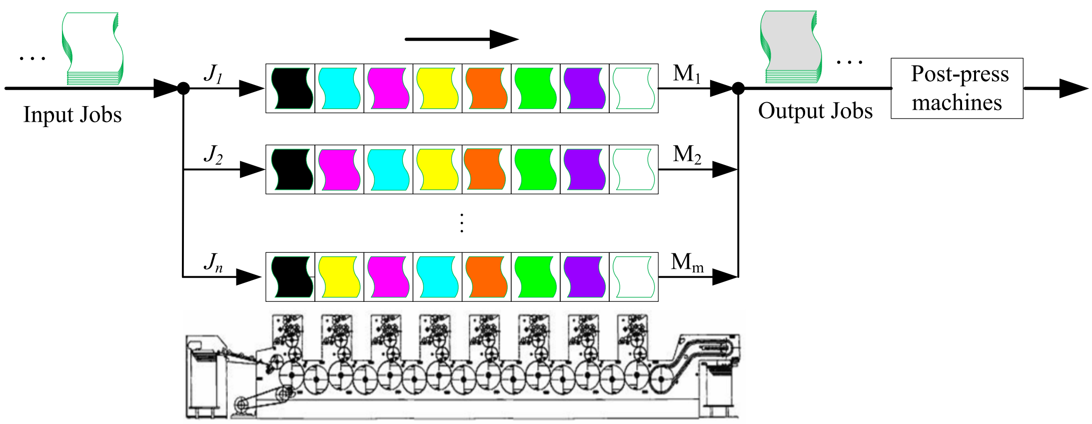


FIGURE 1. The parallel color printing press scheduling framework. n independent printing jobs $(J_1, J_2, \ldots, J_n)$ are printed on m sets of the same 8-color printing presses $(M_1, M_2, \ldots, M_m)$, and the printing time of n jobs is different.

Continuous printing of two jobs with different color sequences requires replacing inks and cleaning ink rollers in the printing production. During this process, it not only wastes a lot of time, but also brings environmental pollution and other issues. Therefore, it is necessary to group jobs with the same color sequences together into a job group for continuous printing. In this paper, a job sequence optimization approach is proposed to alleviate congestion, reduce delay time, and improve the effective utilization of equipment. The objective function for this approach is makespan minimization to improve production management efficiency and reduce production costs.

Indices:

- i Index for jobs in the cell, $i \in \{1, 2, \ldots, n\}$
- j Index for the color sequence job groups, $j \in \{1, 2, \ldots, l\}$
- x Index for the number of colors
- k Index for the printing press in the printing shop, $k \in \{1, 2, \ldots, m\}$

Parameters:

- T_k Working time of printing press k
- T_r Time of replacing ink and cleaning ink rollers
- C_i The printing quantity of job i
- G_j Color sequence job group j
- T_{cd} Time of changing plates and debugging
- $T_{j,k}$ Printing time of job group j on printing press k
- $T_{i,k}$ Printing time of job i on printing press k
- v_k Printing speed of printing press k
- ρ_j Saving time of job group j
- TST Total saving time
- TPT Total printing time
- C_{max} Maximum makespan of job shop

Decision Variables:

$$\text{Min } max\{\sum_{k=1}^{m}\sum_{j=1}^{l} T_{j,k}\} \tag{1}$$

Formula (1) is the objective function. The aim is to schedule all operations to minimize the makespan.

$$X_{i,j} = \begin{cases} 1, & \forall J_i \in G_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$Y_{j,k} = \begin{cases} 1, & \forall G_j \in M_k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$X_{i,j}$ 1 if job i is assigned to color sequence job group j; 0 otherwise

$Y_{j,k}$ 1 if color sequence job group j is assigned to the printing press k; 0 otherwise

Formula (2) indicates whether a printing job belongs to one color sequence job group. Formula (3) indicates that each color sequence job group must and can only select one printing press.

$$T_{i,k} = \frac{C_i}{v_k} \tag{4}$$

$$T_{j,k} = T_r + \sum_{i=1}^{n}\sum_{j=1}^{l}(1 - Y_{j,k}) \cdot \left[T_{cd} + T_{i,k} + (1 - X_{i,j}) \cdot T_r\right] \tag{5}$$

$$T_k = \sum_{j=1}^{l} T_{j,k} \tag{6}$$

$$T_k \geq 0 \quad \forall k \tag{7}$$

$$C_i \geq 0 \quad \forall i \tag{8}$$

$$v_k \geq 0 \quad \forall k \tag{9}$$

Formulae (4) - (9) are used to calculate the working time of printing press.

$$TPT = \sum_{i=1}^{n}\sum_{j=1}^{l}\sum_{k=1}^{m}\left(T_{cd} + T_{i,k} + T_r\right) \tag{10}$$

$$TST = \sum_{j=1}^{l} \rho_j \tag{11}$$

Formula (10) shows that when all jobs are finished, the total printing time can be calculated. Formula (11) is used to calculate the total saving time.

B. NUMERICAL ILLUSTRATION

There is a set of jobs, $J = \{J_1, J_2, \ldots, J_n\}$, and a set of 8-color printing presses in the printing shop, $M = \{M_1, M_2, \ldots, M_m\}$. Here, $\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$ represents the set of M lists. For example, consider a problem with 9 jobs and 3 printing presses, a possible solution is represented as $\pi = \{(J_1, J_4, J_7), (J_2, J_5, J_9), (J_3, J_6, J_8)\}$. In this solution, the operation sequences of jobs in 8-color printing press M_1, M_2, and M_3 are $J_1 - J_4 - J_7$, $J_2 - J_5 - J_9$ and $J_3 - J_6 - J_8$, respectively.

For a numerical illustration of the considered problem, let us assume a PCPSP problem. In a printing workshop, there are two same 8-color printing presses, $m = 2$. The printing workshop has received a batch of printing tasks that require four printing jobs, $n = 4$. Color sequences required for printing jobs are shown in Table 1. Printing equipment debugging time, printing time, and ink replacement time of jobs are shown in Table 2. Suppose $\pi = \{J_1, J_2, J_3, J_4\}$ is the sequence for which we want to find the minimum completion time. Following the first-come first-served (FCFS) rule, when assigning the first job J_1, since both the printing presses are empty, there is no difference in assigning it to printing press M_1 or M_2. So, job J_1 is arbitrarily assigned to printing press M_1 with a completion time of 62 minutes. After scheduling job J_1, the next job in sequence is J_2. The starting time of job J_2 in printing press M_1 is 62 minutes and in printing press M_2 is 0 minute. Therefore, job J_2 is assigned to printing press M_2 as per FCFS rule. This process is repeated for the remaining jobs J_3 and J_4. Based on the FCFS rule, the solution is represented as $\pi_{\text{FCFS}} = \{(J_1, J_4), (J_2, J_3)\}$ and the minimal makespan is 122 minutes.

For assigning jobs to different printing presses, Naderi and Ruiz show that the earliest completion time (ECT) rule gives a better result at a very small computational cost [34]. When assigning these four jobs, if the printing jobs J_1 and J_3 are arranged in a printing press, and the printing jobs J_2 and J_4 are arranged on another printing press, the minimal makespan is 116 minutes. Based on the ECT rule, the scheduling solution is represented as $\pi_{\text{ECT}} = \{(J_1, J_3), (J_2, J_4)\}$, and the earliest completion time of printing workshop can save 6 minutes.

TABLE 1. Color sequences of 4 printing jobs in Figure 1.

Jobs	Color sequence 1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}
J_1	BK	C	M	Y	O	G	V	W
J_2	BK	C	M	Y	G	V		
J_3	BK	M	C	Y	O	G	V	
J_4	BK	M	C	Y	O			

However, as shown in Table 1, we can also find that compared to the printing color sequence in job J_1, job J_2 lacks two colors, O and V. If job J_2 is continued to process after printing job J_1, the operator can continue to printing by separating the printing plate cylinder and form inking roller in the unit 5 and unit 8. Since it's not necessary to replace the ink and clean the ink rollers, the completion time of jobs J_1 and J_2 can be reduced 30 minutes. The same applies to jobs J_3 and J_4. Based on the color sequences comparison (CSC), this solution is represented as $\pi_{\text{CSC}} = \{(J_1, J_2), (J_3, J_4)\}$. Once the scheduling of all the jobs is known, the printing time of any printing press can be calculated. The minimal makespan is 88 minutes, which is clearly smaller than the first two scheduling schemes. A Gantt chart is shown in Figure 2.

TABLE 2. Time for different printing actions.

Printing action	Job J_1	J_2	J_3	J_4
T_{cd}	10	10	10	10
T_i	22	16	14	20
T_r	30	30	30	30

V. JOB SEQUENCE OPTIMIZATION APPROACH

From the illustrated example, the color sequences of printing jobs have the greatest impact on the printing time of the printing presses during the PCPSP. How to schedule printing jobs with the same or similar color sequence together for printing has become a key scientific issue. Therefore, to achieve the goal of minimizing printing time, we propose two job sequence optimization approaches: Moving Interpolation (MI) algorithm and Color Sequence Mapping (CSM) algorithm. We introduce three important definitions in the algorithms.

Definition 1: The set of jobs to be printed is called *cell*.

Definition 2: The first job to be printed after each ink change is called a *sample job*.

Definition 3: Other jobs used to be compared with the color sequence of sample job are called *comparison jobs*.

In job sequence optimization approach, the selection of the *sample job* also has a significant impact on the allocation of subsequent jobs. Therefore, the selection of *sample job* is crucial. After analyzing the characteristics of the printing process, it is concluded that as a *sample job*, the following conditions should be met. (1) There should be the maximum

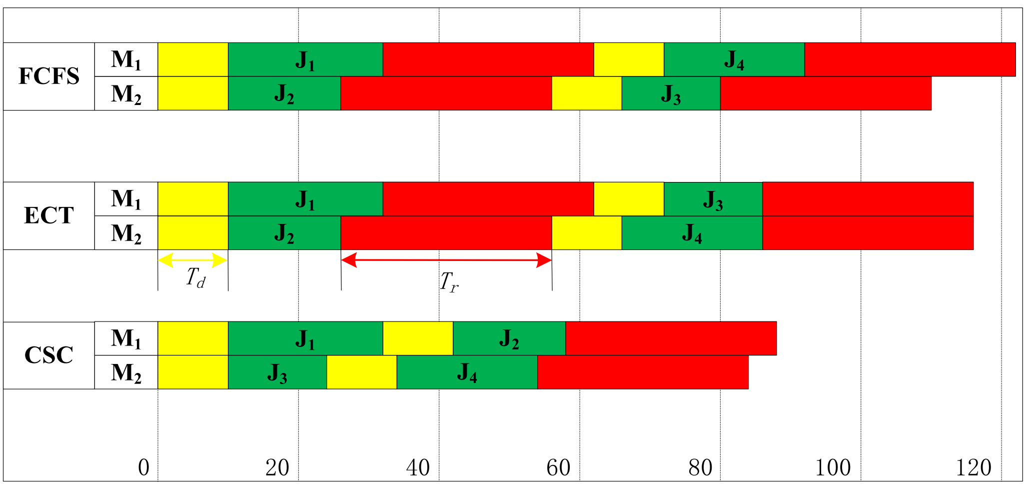


FIGURE 2. The Gantt Chart for the illustrated example. Assign 4 printing jobs to 2 printing machines using three rules: first come first served (FCFS), earliest completion time (ECT), and color sequences comparison (CSC).

number of overprint colors in the *sample job*. (2) The overprint colors of the sample job should be the same or similar to the color sequence of the frequently printed jobs in the printing shop.

A. MOVING INTERPOLATION ALGORITHM

The color sequence interpolation process of the MI algorithm is shown in Figure 3. Depending on the printing characteristics of jobs, the exploitation phase of the MI algorithm is divided into the five following steps:

Step 1: Retrieve jobs to be printed from *cell* and record their quantity as n. Select the job with the largest number of colors from *cell* as the *sample job*. The remaining jobs are considered as *comparison jobs*. The color sequence of the *sample job* is retrieved from the database.

Step 2: According to $(1, 2, \ldots, 8)$, the colors of *sample job* in order are marked. Read the color sequence of each *comparison job* in turn from the database. The colors of *comparison jobs* are marked with the value of *sample job.* According to (123, 234, ..., 678, 78, 8), the colors of *sample job* and *comparison job* are compared. If there are any identical corresponding colors within the group, the comparison continues with the next group of colors. If there is no corresponding color match, insert 0 at the beginning of the current color group and proceed to compare it with the next group of colors. The maximum number of insertion times in the comparison process is K_i,

$$K_i = N_{max} - N_i \tag{12}$$

where, N_{max} is the number of *sample job* colors, N_i is the number of *comparison job* colors. When the *comparison job* reaches the maximum number of insertion times K_i, the interpolation process is completed.

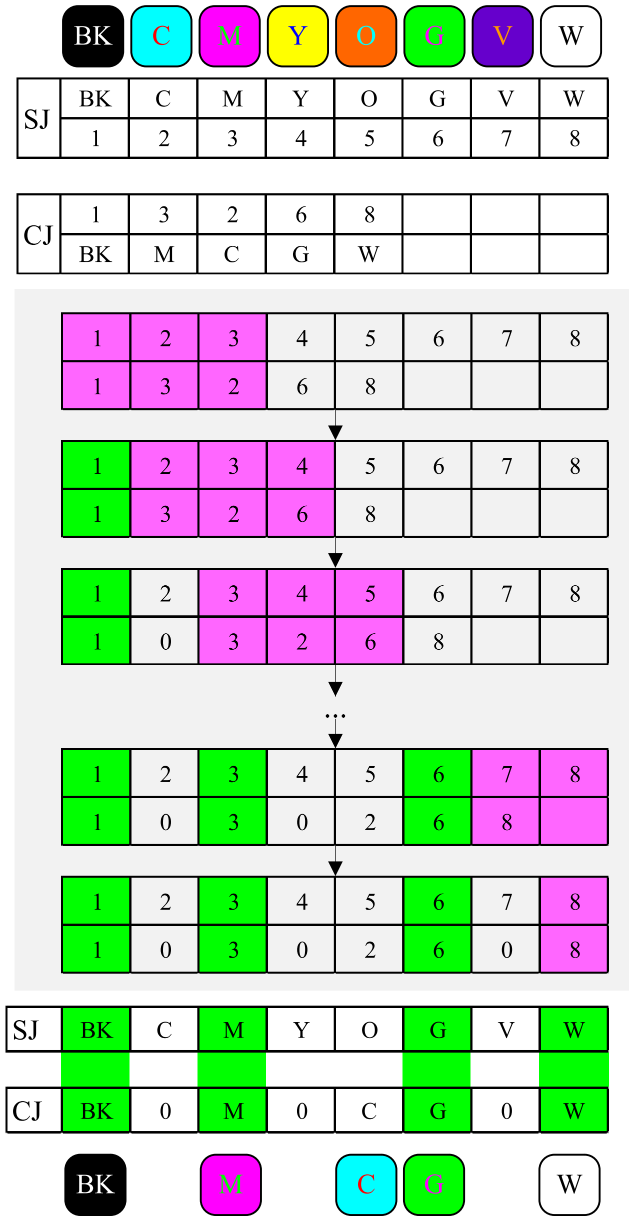
Step 3: According to the color $(1, 2, \ldots, 7, 8)$ compare the color sequence of *sample job* and *comparison job* with 0 value. The number of the same colors are called *Similarity*, which is recorded as R_i. The number of difference colors are called the *Dissimilarity*, which is recorded as D_i

$$D_i = N_i - R_i + K_i \tag{13}$$

Step 4: The comparison jobs with $D_i = 0$ are set as a group, which is marked as a *color sequence job group* G_j.

Step 5: Select the next *sample job* with the largest number of colors from the rest *comparison jobs*. Repeat Steps 2-5 until there is no *comparison job*.

The pseudo-code of **Moving interpolation algorithm** is shown in Algorithm 1.



SJ: Sample Job，CJ: Comparison Job
BK: Black, C: Cyan, M: Magenta, Y: Yellow,
O: Orange，G: Green, V: Violet, W: White

FIGURE 3. Illustration of moving interpolation algorithm.

B. COLOR SEQUENCE MAPPING ALGORITHM

The color sequence mapping process of CSM algorithm is illustrated in Figure 4. Depending on the printing characteristics of jobs, the exploitation phase of CSM algorithm is divided into the five following steps:

Step 1: Retrieve jobs to be printed from *cell* and record their quantity as n. Select the job with the largest number of colors from *cell* as the *sample job*. The remaining jobs are considered as *comparison jobs*. The color sequence of the *sample job* is retrieved from the database.

Step 2: According to $(8, 7, \ldots, 1)$, the colors of *sample job* are marked in order. Retrieved the color sequence of each *comparison job* in turn. The colors of *comparison jobs* are marked with the value of the *sample job.* Map the colors value of *comparison jobs* to *sample job*. Judge whether the color value of *comparison job* is greater than or equal to its standard color sequence coordinates. If $CJ(i) \geq SJ(i)$, $i = i + 1$;

Algorithm 1 Pseudo Code of MI Algorithm

```
Procedure MI algorithm
G = {G1, G2, ..., Gl}
cell=Get color sequence of all current jobs to be printed
[n, ~]=size(cell)
for j = 1 to n
SJ=Randomly get the color sequence of job with Nmax
   from cell
[~, Nmax] = size (SJ)
for i=1 to n
   for e= 1 to Nmax
      CJ (e) = cell (i, e)
      endfor
   Ni = Calculate the number of colors for CJ (e)
   for e = 1 to Nmax
      if Ni < Nmax && e <7 &&colors of three corresponding
groups are different
      CJ=[CJ (1: e) 0 CJ (e + 1:end)])
      Ni = Ni + 1
   elseif Ni < Nmax && e = 7 &&colors of 7th, 8th
corresponding groups are different
      CJ=[CJ (1: e) 0 CJ (e + 1:end)])
      Ni = Ni + 1
   else
      break
   endif
   endfor
      Calculate Ki according to Formula (12)
      Calculate the diversity Di according to Formula (13)
   if Di ==0 then
   CJ (i) is marked as a job group Gj
   Delete CJ (i) from the cell
   endif
endfor
   if cell==[]
      break
   endif
endfor
Return G
```

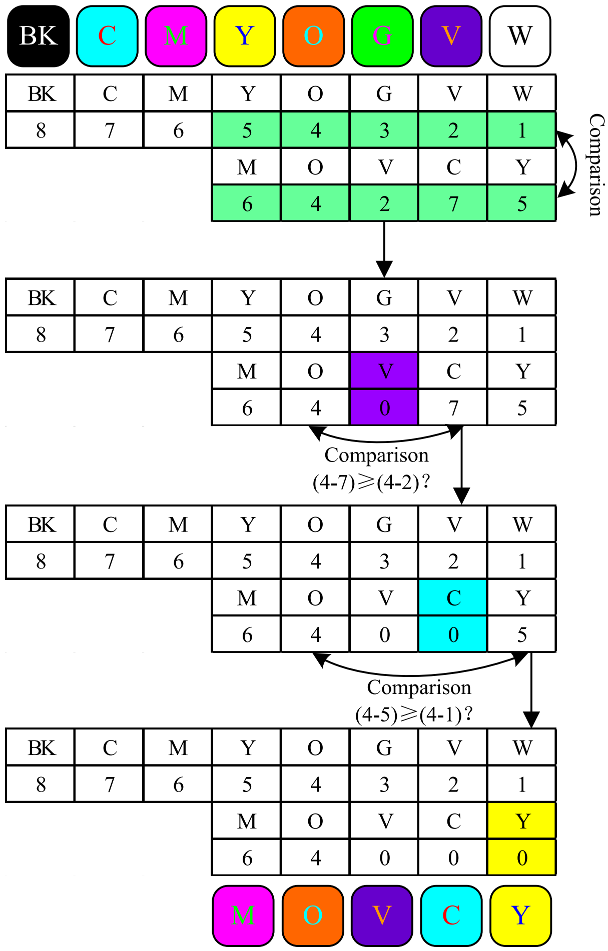


FIGURE 4. Illustration of color sequence mapping algorithm.

otherwise, remark the color $CJ(i) = 0$. This means that the color cannot be printed without changing the ink.

Step 3: Compare the colors of two adjacent values that both are not 0 in the *comparison job* from left to right. Judge whether the difference between the left color sequence number and the right color sequence number is greater than or equal to their color sequence coordinate difference. If $CJ\ (i) - CJ\ (j) < SJ\ (i) - SJ(j)$, remark the color $CJ\ (j) = 0$; else, it will not be handled.

Step 4: Judge whether color can be printed without changing the ink. If $(9 - CJ\ (i)) < i$, remark the color $CJ\ (j) = 0$; else, it will not be handled (such as [0**8 7** 5 4 3]).

Step 5: The number of 0 are the *Dissimilarity*, which is recorded as D_i. The comparison jobs with $D_i = 0$ are set as the same group, which is marked as a *color sequence job group* G_j.

Step 6: Select the next *sample job* with the largest number of colors from the rest *comparison jobs*. Repeat Steps 2-5 until there is no *comparison job*.

The pseudo-code of **Color sequence mapping algorithm** is shown in Algorithm 2.

C. ITERATED GREEDY ALGORITHM

The Iterated Greedy (IG) algorithm for flowshop scheduling problem was first proposed by Ruiz and Stiitzle [35]. As mentioned, IG algorithm has been successfully applied several times to minimise the makespan of PMSP in the past. Therefore, in this paper, IG algorithm is used to schedule the color sequence job group in the printing shop.

In order to balance the load of the printing press and improve the utilization rate of the printing press, this paper uses the following two basic principles. (1) All job groups should be allocated first and then sorted. (2) The shorter the

Algorithm 2 Pseudo Code of CSM Algorithm

Procedure CSM algorithm
$G = \{G_1, G_2, \ldots, G_l\}$
cell=Get color sequence of all current jobs to be printed
[n, ∼]=size(*cell*)
for $j = 1$ to n
SJ=Randomly get the color sequence of job with N_{max} from *cell*
[∼, N_{max}]=size(SJ)
for each CJ(e) in cell(i, e) **do**
  Mark number (8)-(1) the color of CJ according to SJ
  for $e = 1$ to N_{max}
    if CJ(e) >SJ(e)
      CJ(e) =0
    endif
  endfor
  for $e = 1$ to N_{max}
    for $x = 1$ to N_{max}
    if CJ(e) ∼= 0 && CJ(e+x) ∼= 0 && e+x < N_{max} && CJ(e)-CJ(e+x) <SJ(e)-SJ(e+x)
      CJ(e) =0
    endif
    endfor
  endfor
    for $x = 1$ to N_{max}
    if (9-CJ(i))< i
      CJ(e) =0
    endif
  endfor
    Calculate K_i according to Formula (12)
    Calculate the diversity D_i using Formula (13)
  if D_i ==0
    CJ (i) is marked as a job group G_j
    Delete CJ (i) from the *cell*
    endif
  endfor
    if *cell*==[]
      break
    endif
endfor
Return G

running time of the printing press, the higher the priority of obtaining jobs. In the following section, we describe each step of IG algorithms in detail.

Step 1: The accumulated working time of all printing presses are recorded as $T_1, T_2, \ldots, T_m$,and their initial values are set 0.

Step 2: Use formulas (1), (2), (4) and (5) to calculate the time $T_{j,k}$ required for all *color sequence job groups* to be printed, and sort $T_{j,k}$ in decreasing order according to the largest-processing-time-first (LPTF) rule.

Step 3: Schedule the *color sequence job group* with max($T_{j,k}$) to the printing press which has the smallest cumulative working time.

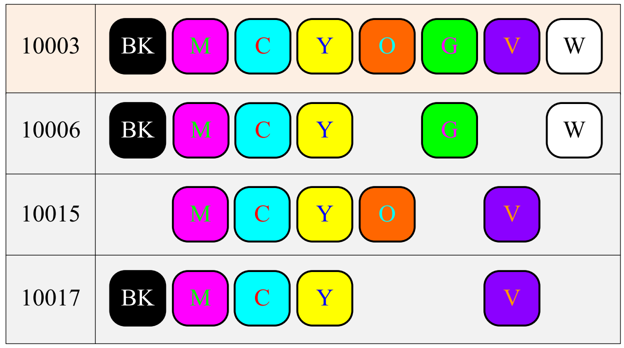


FIGURE 5. The color sequences of job group G_3.

Step 4: Iteratively execute until all *color sequence job groups* are scheduled. Calculate the completion time of each printing press T_k using Formula (3).

The pseudo-code of **Iterated greedy** algorithm is shown in Algorithm 3.

Algorithm 3 Pseudo Code of IG Algorithm

Procedure IG algorithm
$\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$
$T = \{T_1, T_2, \ldots, T_m\}$
$G = \{G_1, G_2, \ldots, G_l\}$
for $j = 1$ to l
Calculate $T_{j,k}$ using Formulae (2), (3), (4) and (5)
G' = Sort l job groups according to their $T_{j,k}$ in decreasing order
endfor
initialize $T_{min}, T_1, T_2, \ldots, T_m \leftarrow 0$
for $j = 1$ to l
  $T_{min} = min(T_1, T_2, \ldots, T_m)$
  if $T_k == T_{min}$
    $T_k = T_k + T_{j,k}$
    $\pi_k = \ldots, G'_j$
  endif
endfor
Return π and T

VI. EXPERIMENT AND DISCUSSION

In this section, three sets of computational experiments are performed to evaluate the performance of the proposed JSOA mechanism. All the experiments are encoded in MATLAB (R2015a) and run on a personal computer with an Intel (R) Core (TM) i7-6700HQ CPU @ 2.60 GHz and 16GB RAM under a windows10 operating system environment.

A. FEASIBILITY AND EFFECTIVENESS OF ALGORITHMS

In order to verify the feasibility and effectiveness of the proposed algorithms. The experimental parameters are set based on the printing press type and customer order type in the printing shop of a pharmaceutical packaging enterprise. There are twenty-two color printing jobs ($n = 22$) and three

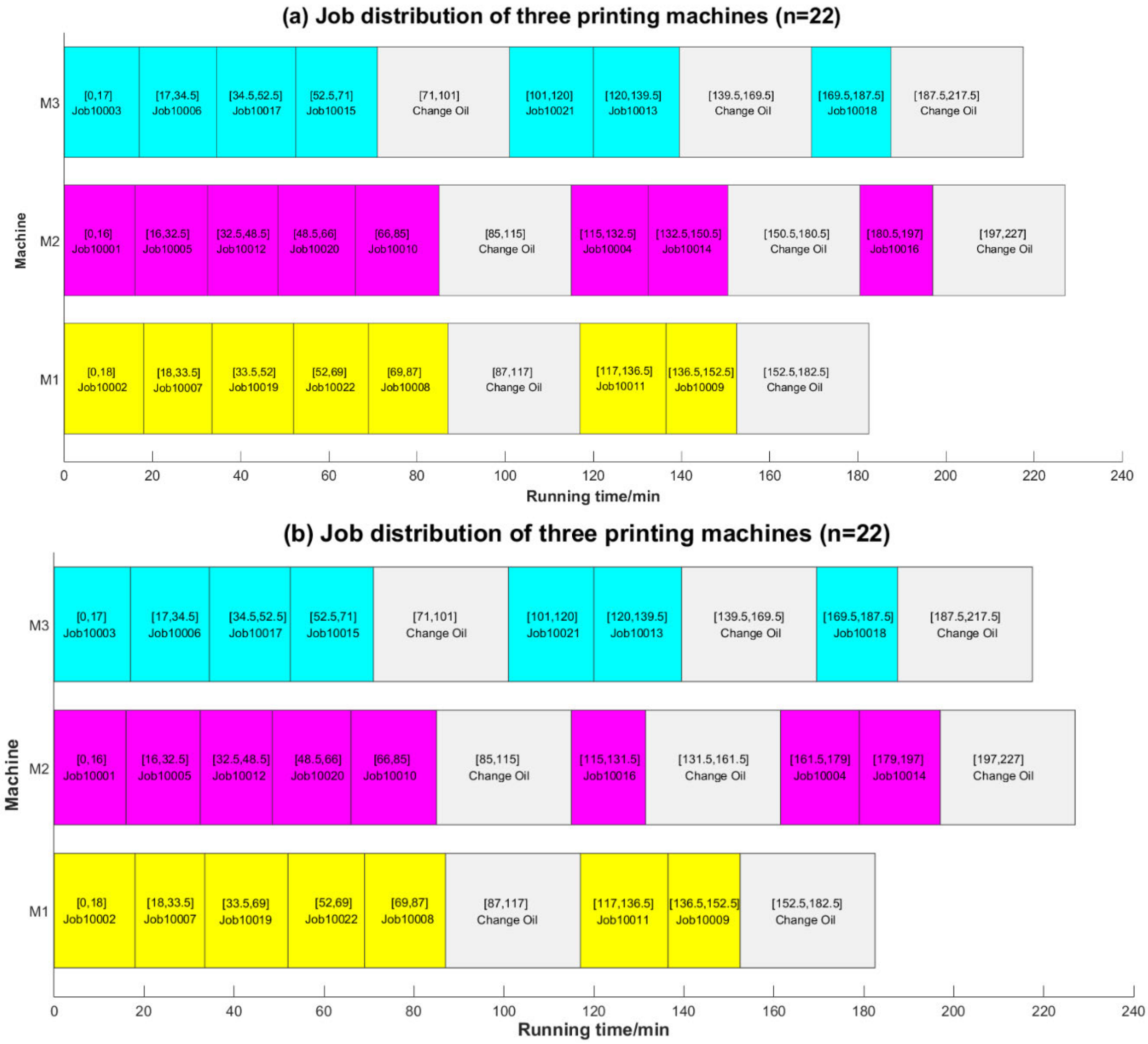


FIGURE 6. Scheduling schemes of job groups ($n = 22$). The Gantt chart of job groups initial scheduling scheme is shown in Figure 6(a). The Gantt chart of final scheduling scheme is obtained by sorting the job groups of each printing press, as shown in Figure 6(b).

8-color printing presses ($m = 3$) for PCPSP. These printing jobs (see Table 4 in Appendix) are designed according to the principles of the color sequence in Section III. The speed of printing press v_k is 200 sheets per minute.

This paper uses the MI algorithm or the CSM algorithm to group and sort the 22 printing jobs in Table 3. The grouping and sorting results of two algorithms are summarized in Table 3. Taking *color sequence job group* G_3 as an example, we introduce the scheduling process of printing jobs. As shown in Figure 5, the color sequence of sample job (No.10003) is {BK, M, C, Y, O, G, V, W}. After completing the sample job, the comparison job (No.10006) is scheduled. In addition to replacing the printing plates of six units (1-4, 6, 8), it is only necessary to open the clutches between the printing plate cylinder and the form ink roller in the 5th and 8th units. The clutches are opened in the 1st, 6th, and 8th units when the comparison job (No.100015) is printed. The clutches are opened in the 5th, 6th, and 8th units when the comparison job (No.100017) is printed. The entire printing process of G_3 does not require to replace ink and clean the ink rollers.

TABLE 3. The results of 22 printing jobs grouping and sorting.

G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
10001	10002	10003	10004	10011	10021	10016	10018
10005	10007	10006	10014	10009	10013		
10012	10019	10015					
10020	10022	10017					
10010	10008						

From the experimental results shown in Table 3, both the MI algorithm and the CSM algorithm have given correct results, which indicates that both algorithms have good feasibility.

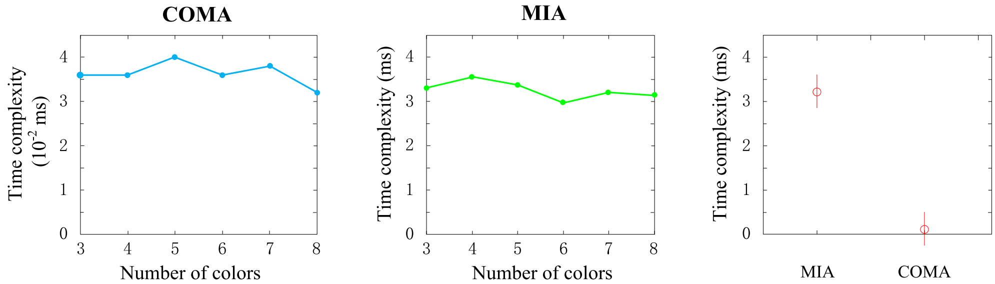


FIGURE 7. Time efficiency of the CSM algorithm and the MI algorithm.

If the MI algorithm and the CSM algorithm are not used on PCPSP, the worst total printing time is calculated by using Formula (10), $TPT = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(T_{cd}+T_{i,k}+T_r\right) =$ 1047min. Assuming that in an ideal scenario, these jobs are assigned to each printing press (i.e., each printing press has the same completion time). The worst maximum completion time of printing shop is $C_{max} = {}^{TPT}\!/_{m} = 349$min.

In this paper, the IG algorithm is utilized to schedule the eight job groups in Table 3 on $\{M_1, M_2, M_3\}$ printing presses in the printing shop. The Gantt chart of job groups initial scheduling scheme is shown in Figure 6(a).

Although the job groups inevitably change the ink during continuous printing, a reasonable sequence can also reduce the frequency of replacing ink and cleaning ink rollers. Therefore, the scheduling scheme of printing shop can be optimized by sorting the job groups for each printing press. The specific operating steps are as follows. Firstly, we compare the color sequences of sample jobs in job groups using the MI algorithm (or CSM algorithm). Then, we sequentially find the job group with the smallest difference (i.e., $min(D_i)$) from the current job group by using IG algorithm. Finally, the color sequence job groups are sorted in non-decreasing order. The Gantt chart of final scheduling scheme is shown in Figure 6(b).

The gray rectangle represents the time of replacing ink and cleaning ink rollers T_r when a job group is finished. Since the processing time of each job must include the time for changing plates and debugging and printing time (i.e., $T_{cd} + T_{i,k}$), we combine the two times to one part. For example, [0, 17] represents the processing time of job 10003 in Figure 6.

According to Formula (6), the completion time of each printing press can be calculated as 217.5min, 227min, 182.5min. This means the maximum makespan of printing shop is $C_{max} = 227$min, which is 122 minutes earlier than if the JSOA were not used. The total reducing time is $TRC = \sum_{j=1}^{n}\delta_j = 4\times 30 + 5\times 30 + 5\times 30 = 420$min, which is over 40% of the TPT.

The traditional printing workshop of the pharmaceutical packaging enterprise follows the FCFS rule for printing, and the printing times of the three presses are 380.5min, 334min, and 332.5min, respectively. The actual maximum makespan is $C_{max} = 380.5$min. Compared with the traditional method, the maximum makespan of our proposed optimization algorithm is saved by 153.5min, and the efficiency is improved by 40.3%. Therefore, we can conclude that the proposed MIA algorithm and the CSM algorithm have a significant impact on PCPSP schemes.

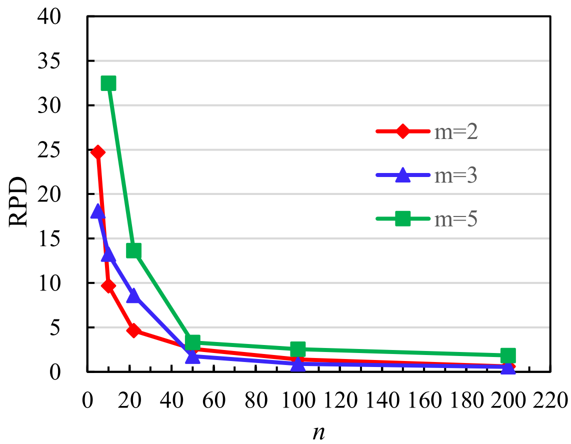


FIGURE 8. RDP for the JSOA with small instances.

B. TIME EFFICIENCY OF MI AND CSM ALGORITHMS

The second experiment is conducted to test the efficiency of the proposed MI and CSM algorithms, this paper designs a comparative experiment. Firstly, we set up an 8-color sample job, $color = \{$BK, C, M, Y, O, G, V, W$\}$. Secondly, these 8 colors are used to generate 6 types of comparison jobs, whose number of colors range from 3 to 8. Three comparison jobs were generated for each type, and their color sequences were randomized. Subsequently, 18 comparison jobs (6×3) are compared with the sample job using MI and CSM algorithms. Finally, the time efficiency of the two algorithms is statistically analyzed.

To avoid accidental errors, the programs of MI and CSM algorithms for each comparison job run three times to obtain an average running time. The average value of three same type jobs is considered as the time efficiency. The computational results are shown in Figure 7 (see Table 5 in Appendix). The experimental results demonstrate that the time efficiency of the MIA ranges from 2.9ms to 3.6ms, with an average time efficiency of 3.27ms. On the other hand, the time efficiency of the CSM algorithm ranges from 0.03ms to 0.04ms, with an

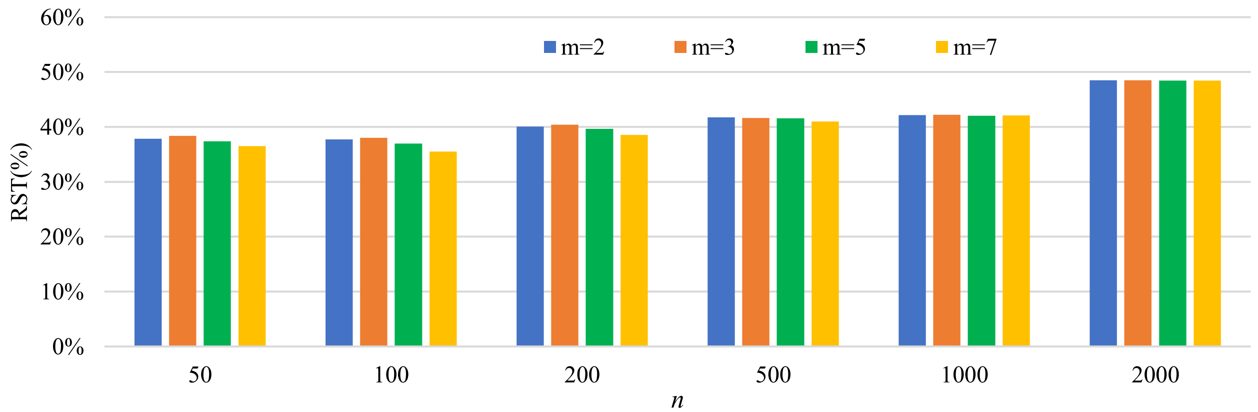


FIGURE 9. Effect of the number of printing jobs changes on the JSOA approach. Figure 9 illustrates the ratio of saving time (RST) under different numbers of printing jobs and printing presses.

average value 0.036ms. It is evident that the time efficiency of the CSM algorithm is much better than the MI algorithm. The primary reason is the number of loop iteration and logical judgment in the MI algorithm. Additionally, the time efficiency is less affected by the number and sequence of colors.

C. EVALUATIONS OF THE PROPOSED ALGORITHMS

Since the JSOA with color setting criteria is addressed for the first time in this paper, there is no standard Hi-Fi printing data set for it. Each problem of $DF|prmu|\sum T_j$ is defined by four variables: the number of printing presses in printing shop, the number of jobs, the number of colors, and printing quantity of every job. Therefore, in order to evaluate the degree to which the final scheduling scheme is close to the optimal solution, we use the proposed JSOA to test two sets of instances. The first set consists of small-sized problems with printing presses m ={2, 3, 5}, and jobs n ={5, 10, 22, 50, 100, 200}. The second set comprises large-sized problems with printing presses m ={2, 3, 5, 7}, and jobs n ={50, 100, 200, 500, 1000, 2000}. The color sequence of jobs is randomly generated, with the number of colors denoted as x = {8, 7, 6, 5, 4, 3}, accounting for 10%, 15%, 20%, 25%, 20% and 10% in the jobs respectively. Based on the actual production experience of small batch printing, the printing quantity of each job is also generated randomly from 1100 to 2000.

The proposed methods are heuristics and are not expected to find the optimum solution. Relative percentage deviation (RPD) over the best solution found in the experiment is considered as performance measure [8], [36].

$$RPD = \frac{Alg_{sol} - Best_{sol}}{Best_{sol}} \times 100 \tag{14}$$

where Alg_{sol} is the minimal makespan which is obtained by the proposed JSOA approach. $Best_{sol}$ is an ideal C_{max} which can be calculated after using the MI algorithm or the CSM algorithm to group and sort all jobs.

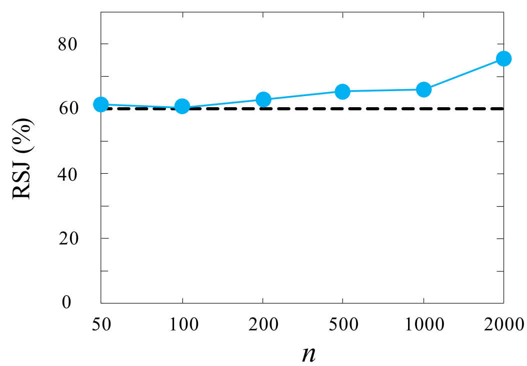


FIGURE 10. RJS under different number of jobs.

We evaluate the RPD of the JSOA approach. Figure 8 depicts the effects of the increase of jobs n on JSOA approach, which also illustrates the effectiveness of the designed color sequence comparison mechanism. As shown in Figure 8, the function $RPD = f(n)$ is convergent and converges to 0. The smaller the value of m, the faster the convergence speed of the function RPD. When the number of printing jobs n is within the range of [5, 50], its RPD convergence speed is the fastest. When $n/m > 20$, the RPD value is less than 3%, which illustrates that the JSOA proposed is very close to the optimal solution in an ideal situation in this paper.

In the large-sized experiment, we study the scalability of the problems solved by the proposed JSOA approach efficiently regarding the parameter n, which represents the numbers of jobs. The ratio of saving time (RST) is an index used to measure JSOA performance.

$$RST = \frac{TST}{Worst_{sol}} \times 100\% \tag{15}$$

where $Worst_{sol}$ represents the worst C_{max} obtained with different sequence color order of any two continuous jobs in printing press.

RST can be calculated using Formulae (12) and (15). Figure 9 illustrates the change of JSOA performance

TABLE 4. Details of printing jobs ($n = 22$).

Job_ID	Color sequence								Rule	Quantity(*sheet*)
	1^{th}	2^{th}	3^{th}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}		
10001	BK	C	M	Y	O	G	V	W	R_1	1200
10005	BK	C	M	Y	O	G			R_1	1300
10012	BK	C	M	Y	W				R_1	1200
10020	C	M	Y	V					R_1	1500
10010	BK	C	Y	O	G				R_2	1800
10018	Y	BK	C						R_2	1600
10003	BK	M	C	Y	O	G	V	W	R_3	1400
10006	BK	M	C	Y	G	W			R_3	1500
10017	BK	M	C	Y	V				R_3	1600
10011	M	C	Y	G	O				R_3	1900
10002	BK	Y	M	C	O	G	V	W	R_4	1600
10007	BK	Y	M	C	V	W			R_4	1100
10019	BK	Y	M	C					R_4	1700
10022	Y	M	C						R_4	1400
10021	M	C	Y	W	G	V			R_5	1800
10015	M	C	Y	O	V				R_5	1700
10009	BK	M	C	Y	G	O			R_5	1200
10013	M	C	Y	W	G				R_5	1900
10004	W	C	M	Y	O	V	BK		R_6	1500
10014	C	M	Y	O	V	BK			R_6	1600
10008	Y	M	C	O	V	W			R_6	1600
10016	C	M	Y	O	G	V	BK		R_6	1300

TABLE 5. Time efficiency of the proposed MI and CSM algorithms.

Algorithm	Number of colors						Average time efficiency (ms)
	3	4	5	6	7	8	
MI	3.31	3.55	3.37	2.98	3.22	3.16	**3.27**
COM	0.036	0.036	0.04	0.036	0.038	0.032	**0.036**

under different numbers of printing jobs. In general, its performance improves with the increase of printing jobs and gradually stabilizes. When the number of jobs $n \geq 50$, all RST values exceed 35%. Almost all RST values exceed 40% when the number of printing jobs $n \geq 200$. This illustrates that the proposed JSOA can significantly reduce the makespan of the printing shop.

TABLE 6. Detailed results of the JSOA approach for instances.

	n	$Worst_{sol}$	$Best_{sol}$	JSOA	RPD	RST
m=2	5	117	87	108.5	24.71	7.26%
	10	235.5	175.5	192.5	9.69	18.26%
	22	523.5	313.5	328	4.63	37.34%
	50	1180	715	733.5	2.37	37.84%
	100	2365.25	1450.25	1470.5	1.40	37.71%
	200	4711.25	2790.75	2823	0.61	40.08%
	500	11835.75	6885.75	6892	0.09	41.77%
	1000	23679.25	13674.5	13695	0.15	42.16%
	2000	47356.5	24376.5	24377	0.00	48.52%
m=3	5	78	58	68.5	18.10	12.18%
	10	157	117	132.5	13.25	15.61%
	22	349	209	227	8.61	34.96%
	50	786.67	419.67	485	1.75	38.35%
	100	1576.83	966.83	975.5	0.90	38.01%
	200	3140.8	1860.5	1871	0.56	40.43%
	500	7890.5	4590.5	4605	0.32	41.64%
	1000	15786.17	9116.17	9122.5	0.07	42.21%
	2000	31571	16251	16254	0.02	48.52%
m=5	10	94.2	70.2	93	32.48	41.40%
	22	209.4	125.4	142.5	13.64	31.95%
	50	472	286	295.5	3.32	44.81%
	100	946.1	580.1	595	2.57	37.39%
	200	1884.5	1116.3	1137	1.85	39.67%
	500	4734.3	2754.3	2767	0.46	41.55%
	1000	9471.7	5469.7	5489	0.35	42.05%
	2000	18942.6	9750.6	9768.5	0.18	48.43%
m=7	50	337.14	179.86	214	21.71	36.53%
	100	674.44	414.36	435	7.94	35.50%
	200	1346.07	797.36	827.5	5.49	38.52%
	500	3381.64	1967.36	1995	1.95	41.01%
	1000	6765.5	3906.93	3918	0.39	42.09%
	2000	13530.43	6964.71	6973	0.13	48.46%

We compare our method with the scheduling results of a simulated annealing algorithm proposed by Schuurman and Van Vuuren in the literature [22]. The condition is selecting 5 printing presses to print a large number of printing jobs (ours $n = 2000$, Schuurman and Van Vuuren [22] $n = 145$). The ratio of saving time (RST) of Schuurman et al. is 47.85%, and the RST of our method is 48.3%. Besides, the running time of simulated annealing algorithm needs 73903s, while our algorithm only needs less than 9s. It further demonstrates the efficiency of our proposed algorithm.

To test the economic performance of the JSOA, we calculate the ratio of special jobs (RSJ) with the same or similar color sequence to all jobs.

$$RSJ = \frac{n_s}{n} \times 100\% \tag{16}$$

$$n_s = \frac{m \cdot (Worst_{sol} - Best_{sol})}{T_r} \tag{17}$$

where n_s represents the number of jobs with the same color sequence.

Due to these special jobs do not require to replace ink and clean ink rollers during printing, the special jobs are beneficial for reducing total makespan of the printing shop. Figure 10 shows that the RSJ increases with the number of jobs n, and all RJS values are higher than 60%. The reason is that as the number of jobs increases, there are more jobs with the same or similar color sequence. It illustrates that the proposed JSOA can save a significant amount of production costs for the enterprise.

In summary, the performance of the proposed JSOA's approach can be enhanced by increasing the number of jobs, but it is less affected by the number of printing presses. However, the unlimited increase makes no sense in improving the performance further and imposes computational costs. Since a large number of volatile organic compounds (VOC) present in ink and ink cleaning agents, the proposed JSOA not only reduces production costs, but also helps to protect the environment and reduce VOC emissions.

VII. CONCLUSION AND FUTURE WORK

In this paper, a job sequence optimization mechanism for PCPSP is proposed, which aims to minimise the maximum makespan of printing shop. We present two solutions to solve the problem with different computational effort and processing methods. There are two optimization approaches combined with the Iterated Greedy algorithm: a moving interpolation algorithm and a Color Sequence Mapping algorithm. To evaluate the algorithmic performance, three sets of computational experiments are performed on 2000 problem instances. The experiments demonstrate that the proposed JSOA provides a high-quality solution in PCPSP, and exhibits good feasibility and effectiveness. Particularly for large-sized problems involving multiple printing presses, the proposed approach not only improves production efficiency and reduces production costs but also helps protect the environment and reduce VOC emissions.

Besides the proposed PCPSP in the printing shop, some interesting topics are left for further researches on two aspects:

(1) The debugging time can be set separately, rather than assuming that it is within the processing time, which makes it closer to the real-world production.

(2) Exploring multi-objective dynamic scheduling problem in printing workshop. Taking the minimal makespan and the total order delay (or ahead of schedule) as the multi-objectives, this paper discusses the robustness and efficiency of dynamic disturbance events such as emergency order insertion in the printing workshop, print presses failure, order rework, periodic scheduling on JSOA.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Instances at https://github.com/Pikelin/Parallel-scheduling-problem-in-printing-manufacturing-systems.

APPENDIX

THE DETAILED EXPERIMENTAL RESULTS

See Tables 4–6.

REFERENCES

[1] W. T. Lunardi, E. G. Birgin, P. Laborie, D. P. Ronconi, and H. Voos, "Mixed integer linear programming and constraint programming models for the online printing shop scheduling problem," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105020, doi: 10.1016/j.cor.2020.105020.

[2] J. C. Chen, T.-L. Chen, and H. Harianto, "Capacity planning for packaging industry," *J. Manuf. Syst.*, vol. 42, pp. 153–169, Jan. 2017, doi: 10.1016/j.jmsy.2016.12.007.

[3] I.-B. Park and J. Park, "Scalable scheduling of semiconductor packaging facilities using deep reinforcement learning," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 3518–3531, Jun. 2021, doi: 10.1109/TCYB.2021.3128075.

[4] H. Li, S. Zhou, R. Luo, and B. Du, "Operation chain model of reconfigurable printing manufacturing system based on stochastic process algebra," *IEEE Access*, vol. 7, pp. 127475–127485, 2019, doi: 10.1109/ACCESS.2019.2939072.

[5] M. Iori, A. Locatelli, and M. Locatelli, "A GRASP for a real-world scheduling problem with unrelated parallel print machines and sequence-dependent setup times," *Int. J. Prod. Res.*, vol. 61, no. 21, pp. 7367–7385, Nov. 2023, doi: 10.1080/00207543.2022.2148768.

[6] Z. Pan, D. Lei, and L. Wang, "A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5051–5063, Jun. 2022, doi: 10.1109/TCYB.2020.3026571.

[7] S. Maecker and L. Shen, "Solving parallel machine problems with delivery times and tardiness objectives," *Ann. Oper. Res.*, vol. 285, nos. 1–2, pp. 315–334, Feb. 2020, doi: 10.1007/s10479-019-03267-2.

[8] R. Ruiz, Q.-K. Pan, and B. Naderi, "Iterated greedy methods for the distributed permutation flowshop scheduling problem," *Omega*, vol. 83, pp. 213–222, Mar. 2019, doi: 10.1016/j.omega.2018.03.004.

[9] D. Mecler, V. Abu-Marrul, R. Martinelli, and A. Hoff, "Iterated greedy algorithms for a complex parallel machine scheduling problem," *Eur. J. Oper. Res.*, vol. 300, no. 2, pp. 545–560, Jul. 2022, doi: 10.1016/j.ejor.2021.08.005.

[10] D. Lei and M. Liu, "An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance," *Comput. Ind. Eng.*, vol. 141, Mar. 2020, Art. no. 106320, doi: 10.1016/j.cie.2020.106320.

[11] E. Çanakoğlu and İ. Muter, "Identical parallel machine scheduling with discrete additional resource and an application in audit scheduling," *Int. J. Prod. Res.*, vol. 59, no. 17, pp. 5321–5336, Sep. 2021, doi: 10.1080/00207543.2020.1777481.

[12] J. Adan, "A hybrid genetic algorithm for parallel machine scheduling with setup times," *J. Intell. Manuf.*, vol. 33, no. 7, pp. 2059–2073, Oct. 2022, doi: 10.1007/s10845-022-01959-4.

[13] A. Zandi, R. Ramezanian, and L. Monplaisir, "Green parallel machines scheduling problem: A bi-objective model and a heuristic algorithm to obtain Pareto frontier," *J. Oper. Res. Soc.*, vol. 71, no. 6, pp. 967–978, Jun. 2020, doi: 10.1080/01605682.2019.1595190.

[14] X. Wu and A. Che, "A memetic differential evolution algorithm for energy-efficient parallel machine scheduling," *Omega*, vol. 82, pp. 155–165, Jan. 2019, doi: 10.1016/j.omega.2018.01.001.

[15] R. Chen, W. Li, and H. Yang, "A deep reinforcement learning framework based on an attention mechanism and disjunctive graph embedding for the job-shop scheduling problem," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1322–1331, Feb. 2023, doi: 10.1109/TII.2022.3167380.

[16] E. Vallada, F. Villa, and L. Fanjul-Peyro, "Enriched metaheuristics for the resource constrained unrelated parallel machine scheduling problem," *Comput. Operations Res.*, vol. 111, pp. 415–424, Nov. 2019, doi: 10.1016/j.cor.2019.07.016.

[17] R. F. Tavares-Neto and M. S. Nagano, "An iterated greedy approach to integrate production by multiple parallel machines and distribution by a single capacitated vehicle," *Swarm Evol. Comput.*, vol. 44, pp. 612–621, Feb. 2019, doi: 10.1016/j.swevo.2018.08.001.

[18] V. Fernandez-Viagas, J. M. S. Valente, and J. M. Framinan, "Iterated-greedy-based algorithms with beam search initialization for the permutation flowshop to minimise total tardiness," *Expert Syst. Appl.*, vol. 94, pp. 58–69, Mar. 2018, doi: 10.1016/j.eswa.2017.10.050.

[19] Z. Zhao, M. Zhou, and S. Liu, "Iterated greedy algorithms for flow-shop scheduling problems: A tutorial," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1941–1959, Jul. 2022, doi: 10.1109/TASE.2021.3062994.

[20] C. H. Lee, "A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness," *Int. J. Prod. Res.*, vol. 56, no. 6, pp. 2292–2308, 2018, doi: 10.1080/00207543.2017.1374571.

[21] A. P. Burger, C. G. Jacobs, J. H. van Vuuren, and S. E. Visagie, "Scheduling multi-colour print jobs with sequence-dependent setup times," *J. Scheduling*, vol. 18, no. 2, pp. 131–145, Apr. 2015, doi: 10.1007/s10951-014-0400-2.

[22] J. Schuurman and J. H. Van Vuuren, "Scheduling sequence-dependent colour printing jobs," *South Afr. J. Ind. Eng.*, vol. 27, no. 2, pp. 43–59, Aug. 2016, doi: 10.7166/27-2-1119.

[23] W. T. Lunardi, E. G. Birgin, D. P. Ronconi, and H. Voos, "Metaheuristics for the online printing shop scheduling problem," *Eur. J. Oper. Res.*, vol. 293, no. 2, pp. 419–441, Sep. 2021, doi: 10.1016/j.ejor.2020.12.021.

[24] M. Mostajabdaveh, F. S. Salman, and N. Tahmasbi, "Two dimensional guillotine cutting stock and scheduling problem in printing industry," *Comput. Operations Res.*, vol. 148, Dec. 2022, Art. no. 106014, doi: 10.1016/j.cor.2022.106014.

[25] D. Tuyttens and A. Vandaele, "Towards an efficient resolution of printing problems," *Discrete Optim.*, vol. 14, pp. 126–146, Nov. 2014, doi: 10.1016/j.disopt.2014.09.001.

[26] T. Mütze, "Scheduling with few changes," *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 37–50, Jul. 2014, doi: 10.1016/j.ejor.2013.11.011.

[27] J. Leng, X. Wang, S. Wu, C. Jin, M. Tang, R. Liu, A. Vogl, and H. Liu, "A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems," *Int. J. Prod. Res.*, vol. 61, no. 15, pp. 5156–5175, Aug. 2023, doi: 10.1080/00207543.2022.2098871.

[28] R. Zhang, "Environment-aware production scheduling for paint shops in automobile manufacturing: A multi-objective optimization approach," *Int. J. Environ. Res. Public Health*, vol. 15, no. 1, p. 32, Dec. 2017, doi: 10.3390/ijerph15010032.

[29] F. Winter and N. Musliu, "A large neighborhood search approach for the paint shop scheduling problem," *J. Scheduling*, vol. 25, no. 4, pp. 453–475, Aug. 2022, doi: 10.1007/s10951-021-00713-7.

[30] A. Sadeghieh, M. Dehghanbaghi, A. Dabbaghi, and S. Barak, "A genetic algorithm based grey goal programming (G3) approach for parts supplier evaluation and selection," *Int. J. Prod. Res.*, vol. 50, no. 16, pp. 4612–4630, Aug. 2012, doi: 10.1080/00207543.2011.616233.

[31] H. Ma, C. Sun, J. Wang, Z. Yang, and H. Zhou, "Multisystem optimization for an integrated production scheduling with resource saving problem in textile printing and dyeing," *Complexity*, vol. 2020, pp. 1–14, Nov. 2020, doi: 10.1155/2020/8853735.

[32] H. Sun, S. Fan, X. Shao, and J. Zhou, "A colour-batching problem using selectivity banks in automobile paint shops," *Int. J. Prod. Res.*, vol. 53, no. 4, pp. 1124–1142, Feb. 2015, doi: 10.1080/00207543.2014.948970.

[33] D. Li, J. Wang, R. Qiang, and R. Chiong, "A hybrid differential evolution algorithm for parallel machine scheduling of lace dyeing considering colour families, sequence-dependent setup and machine eligibility," *Int. J. Prod. Res.*, vol. 59, no. 9, pp. 2722–2738, May 2021, doi: 10.1080/00207543.2020.1740341.

[34] B. Naderi and R. Ruiz, "The distributed permutation flowshop scheduling problem," *Comput. Operations Res.*, vol. 37, no. 4, pp. 754–768, Apr. 2010, doi: 10.1016/j.cor.2009.06.019.

[35] R. Ruiz and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 2033–2049, Mar. 2007, doi: 10.1016/j.ejor.2005.12.009.

[36] A. Khare and S. Agrawal, "Effective heuristics and metaheuristics to minimise total tardiness for the distributed permutation flowshop scheduling problem," *Int. J. Prod. Res.*, vol. 59, no. 23, pp. 7266–7282, Dec. 2021, doi: 10.1080/00207543.2020.1837982.

HUAILIN LI received the M.S. and Ph.D. degrees in mechanical engineering from the Xi'an University of Technology, Xi'an, China, in 2015 and 2020, respectively. He is currently an Associate Professor with the Faculty of Printing, Packaging and Digital Media, Xi'an University of Technology. His research interests include performance modeling and evaluation for printing manufacturing systems and printing process control.

YINGYING ZHENG received the B.S. degree from the Xi'an University of Technology, Xi'an, China, in 2023, where she is currently pursuing the M.S. degree in light industry technology and engineering with the Faculty of Printing, Packaging and Digital Media. Her research interests include printing process control and multispectral imaging systems.

BANGYONG SUN received the B.S. and Ph.D. degrees in printing engineering and mechanical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2013, respectively. He is currently a Full Professor with the Faculty of Printing, Packaging and Digital Media, Xi'an University of Technology. His research interests include computer vision, machine learning, and multispectral imaging systems.

BIN DU received the M.S. and Ph.D. degrees in mechanical engineering from the Xi'an University of Technology, Xi'an, China, in 2010 and 2016, respectively. He is currently an Associate Professor with the Faculty of Printing, Packaging and Digital Media, Xi'an University of Technology. His research interests include printing electronic materials and printing manufacturing systems.