Online no-wait scheduling of leather workshop supply chain based on particle swarm optimization

Chengjie Huang[1], Xiaohua Zhou[1], Deshuai Hou[2,*]

1. School of Economics and Business Administration, Chongqing University Chongqing 400030, China

2. School of Accounting, Capital University of Economics and Business Beijing 100072, China

115105725@qq.com

ABSTRACT. *Considering the advantages of the partcile swarm optimization (PSO), this paper probes deep into the improvement of the traditional PSO algorithm and its application in leather workshop scheduling. Firstly, the online scheduling of no-wait supply chain was described in details, while improving the PSO algorithm. On this basis, the author proposed an online no-wait scheduling algorithm based on the improved PSO for leather workshop supply chain. After that, the proposed algorithm was used to schedule an example leather workshop. The results show that our algorithm can find the optimal processing plan with a small swarm and through a limited number of iterations, despite the huge amount of orders..*

RÉSUMÉ. *Etant donné que les avantages de l'optimisation par essaims particulaires (PSO, le sigle de « partcile swarm optimization » en anglais), cet article approfondit l'amélioration de l'algorithme de PSO traditionnel et son application dans la planification des ateliers de maroquinerie. Tout d'abord, la planification en ligne de la chaîne d'approvisionnement sans attente a été décrite en détail, tout en améliorant l'algorithme de PSO. Dans ce contexte, l'auteur a proposé un algorithme de planification en ligne sans attente basé sur l'amélioration de PSO de la chaîne d'approvisionnement pour les ateliers de maroquinerie. Après cela, l'algorithme proposé a été utilisé pour programmer un exemple d'atelier de maroquinerie. Les résultats montrent que notre algorithme peut trouver le plan de traitement optimal avec un petit essaim et par un nombre limité d'itérations malgré le nombre considérable de commandes.*

KEYWORDS: *particle swarm optimization (PSO), supply chain, leather workshop, no-wait scheduling.*

MOTS-CLÉS: *optimisation par essaims particulaires (PSO), chaîne d’approvisionnement, atelier de maroquinerie, planification sans attente.*

DOI:10.3166/JESA.51.153-167

1. Introduction

The leather workshop supply chain is the core of the entire leather manufacturing system. The quality and efficiency of leather products are inconceivable without the effective scheduling of the supply chain. However, the scheduling of leather workshop supply chain is a non-deterministic polynomial-time (NP) hard problem, which is difficult to solve with traditional optimization methods, not to mention that most of these methods are purely theoretical.

Recent years has seen the application of swarm intelligence algorithms in supply chain scheduling. Inspired by the swarm behavior of gregarious creatures like ants, wild geese and fish, the swarm intelligence is a new evolutionary computing technology mimicking the bio-social system, which simulates the incalculable group behavior using the local information. The swarm intelligence provides a new way to find the complex solutions to distributed problems, eliminating the need for centralized control or the global model (Zheng and Wu, 2001). Despite the late start, the research on swarm intelligence has yielded fruitful results, including two popular swarm intelligence algorithms: the ant colony optimization (ACO) and the particle swarm optimization (PSO).

The PSO, as an optimization algorithm, has its root in the swarm intelligence theory of Kennedy and Eberhart (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995). Taking the candidate solutions in a population as particles with no volume or mass, the algorithm moves these particles around in the search space according to simple mathematical formulae over the particle's position and velocity. The movement of each particle is affected by its best-known local position and also guided toward the best-known global position, which are updated as better positions are found by other particles. In this way, the swarm is guided toward the optimal solutions. The PSO not only preserves the swarm intelligence of traditional evolutionary algorithms, but also boasts excellent optimization features. As a result, the PSO algorithm has been widely adopted for multi-objective optimization, biological system modeling, artificial neural networks, signal processing, decision support and data classification (Gu *et al*., 2005).

Over the years, the PSO algorithm has been directly adopted or modified for continuous optimization problems. For example, Gu *et al.* (2005) proposes a PSO algorithm for flexible job-shop scheduling, and proves that the algorithm outperforms the genetic algorithm through experiments; in this algorithm, the dimension of each particle is twice the total number of processes, while the process sequence and the machine number of each process are expressed as two sequences of natural numbers. Scholars design the heuristic rule of the smallest position value (SPV) for particle encoding, and introduces the rule to the PSO scheduling of continuous optimization problems (Tasgetiren *et al*., 2004; Tasgetiren *et al*., 2004; Tasgetiren *et al*., 2007). Leticia *et al*. (2004) put forward an evolutionary algorithm to solve the random key coding in scheduling, which relies on a dynamic mutation operator to ensure the swarm diversity, and minimized the total delay of a single-machine scheduling problem with 40 or 50 jobs using the proposed algorithm. Xia *et al*. (2004) encoded particles as integers, modified the particle position and velocity by the rounding method and performed local search through simulated annealing; this integrated strategy achieved better results than the genetic algorithm and the tabu search in various job-shop scheduling problems. Liao *et al*. (2007) provides an improved discrete algorithm for flow shop scheduling, which redefines the particle velocity and its update rule, and verifies that the algorithm outshines the continuous algorithm and two genetic algorithms. Pan *et al*. (2005; 2007) created a discrete position-update PSO algorithm for no-wait flow shop scheduling with the minimal makespan, and improved its performance in view of the features of the problem.

In recent years, the no-wait scheduling problem has arisen in many industries, such as the hot metal roiling industry, chemical industry, pharmaceutical industry, food processing industry and leather processing industry. As a result, more and more scholars have turned their attention to the no-wait scheduling problem. For instance, Mascis *et al*. (2003) implemented no-wait scheduling of a flow shop using a selective graph formula and through heuristics and branch procedures. Schuster *et al*. (2007) proposed two local search algorithms after decomposing the no-wait flow shop scheduling. In general, the existing studies on no-wait scheduling mainly concentrate on flow shops, failing to address job-shops or online scheduling problems.

In light of the above, this paper probes deep into the improvement of the traditional PSO algorithm and its application in leather workshop scheduling. Firstly, the online scheduling of no-wait supply chain was described in details, while improving the PSO algorithm. On this basis, the author proposed an online no-wait scheduling algorithm based on the improved PSO for leather workshop supply chain. After that, the proposed algorithm was used to schedule an example leather workshop. The results show that our algorithm can find the optimal processing plan with a small swarm and through a limited number of iterations, despite the huge amount of orders.

2. Leather workshop supply chain scheduling problem

The leather workshop supply chain scheduling problem, a typical NP-hard problem, aims to minimize the make span of leather workpieces on a given number of machines, provided that all workpieces are processed on the machines in the same order, the processing sequence of the workpieces is set in advance, and the processing time of each workpiece on each machine is already known. The traditional solutions to this problem include exhaustive method, integer programming method as well as branch and bound method. However, these methods are only applicable to small-scale leather workshops. In recent years, many heuristic algorithms and swarm intelligence methods have emerged, ranging from simulate annealing, tabu search, genetic algorithm to the PSO algorithm.

Supply chain scheduling is fundamental to leather workshop management, especially in the era of online commerce. Nowadays, the customer can place an order at any time online. Upon receiving the order, the manufacturer has to arrange the production according to the order urgency and deliver the final product to the customer as quickly as possible. The arrangement should not only optimize the workpiece processing sequence, but also satisfy the customer’s needs in the online orders. Before the workpieces enter the production system, the arrival time, processing time and number of workpieces are all unknown. The real-time feature, coupled with the fuzzy information, adds to the difficulty of the scheduling task.

In our research, the leather workshop supply chain requires real-time scheduling of the online orders. It should be noted that the leather should be processed in a specified time interval. Otherwise, the product will not have the required hardness, color or elasticity. The leather workshop supply chain scheduling problem differs from the traditional job-shop scheduling problem in the following aspects. Firstly, the time to transfer workpieces from one machine to another is negligible because the transfer is completed by robot; secondly, each workpiece must be processed in a specified time interval; thirdly, new workpieces may enter the production system at any time, requiring real-time scheduling of the entire production sequence; fourthly, there is the "no-wait" constraint: a workpiece should be transferred to the next machine immediately after completing the process on the current machine, unless it is being transferred between two machines.

It can be seen that no rescheduling is allowed for any workpiece that has been already scheduled. In other words, the scheduling of new workpieces should not affect the processing of the scheduled jobs. Therefore, this research needs to find a way to arrange the processing sequence of the current workpieces reasonably, and insert the new workpieces into the sequence without changing the schedule of the existing workpieces, such that all workpieces could be processed in time. If there is only one new workpiece, the insertion and adjustment can be determined directly by the traditional algorithm. However, the direct method is no longer feasible when there are many new workpieces. Therefore, this paper resorts to the PSO algorithm to solve the insertion of numerous new workpieces in a short time.

3. Improved PSO algorithm

The mathematical description of the PSO algorithm is given as follows. For N particles in a given *n*-dimensional search space, let $\mathrm{X_i} = (\mathrm{X_{i1}}, \mathrm{X_{i2}}, \dots, \mathrm{X_{in}})$ and $\mathrm{V_i} = (\mathrm{V_{i1}}, \mathrm{V_{i2}}, \dots, \mathrm{V_{in}})$ be the position and velocity of each particle, respectively, $\mathrm{P_i} = (\mathrm{P_{i1}}, \mathrm{P_{i2}}, \dots, \mathrm{P_{in}})$ and $\mathrm{P_g} = \left(\mathrm{P_{g1}}, \mathrm{P_{g2}}, \dots, \mathrm{P_{gn}}\right)$ be the set of known positions of particle i and all the particles in the swarm, respectively, and $\mathrm{Fit_i}$ be the fitness of each particle relative to the optimal value of the objective function. Then, the best-known position of particle i and that of the swarm can be selected from the respective position set and denoted as $\mathrm{P_{best}}$ and $\mathrm{G_{best}}$, respectively. Then, the velocity and position of each particle can be updated in each iteration by the following equations:

$$V_{id}(t+1) = WV_{id}(t) + c_1 r_1\big(P_{id}(t) - X_{id}(t)\big) + c_2 r_2\left(P_{gd}(t) - X_{id}(t)\right) \tag{1}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{2}$$

where d is the dimension of the particle; w is the inertia weight; $\mathrm{c_1}$ and $\mathrm{c_2}$ are acceleration coefficients that adjust the maximum step size towards the $\mathrm{P_{ibest}}$ and

P_{gbest}, respectively; V_i is the velocity of particle i. Note that the particle dimension should not be too large or too small; otherwise, the particle may collide into or fly over the target area. The two acceleration coefficients are positive constants, whose values should be selected properly to prevent slow convergence or local optimum trap. The particle velocity is a random number in the interval [0, 1] and should fall below the maximum velocity V_{max}, which determines the maximum movement distance of particles per cycle.

There are three parts on the right side of equation (1): the inertia part, the cognition part and the social part. The inertia part refers to the inertia of the previous behavior of the particle, i.e. the velocity before the update; the cognition part; the cognition part describes how the information of the particle itself affects the next movement; the social part reflects how the information sharing and cooperation between particles affect the next movement of the particle. The latter two parts jointly affect the velocity update. Equation (2) shows that the particle position update is the combined result of the current position vector and the updated velocity vector.

In our research, the original PSO algorithm must be improved considering the real-time scheduling demand and the fuzzy information of online orders. Let L be the number of workpieces to be processed. Then, an L-dimensional vector was generated to represent a particle in the search space, with each element being one of the L workpieces. Thus, the dimensions of the particle equal the number of workpieces. Then, the workpieces were arranged by the size of their corresponding vector elements. The smaller the vector element, the earlier the corresponding workpiece in the processing sequence. If a machining system contains 6 workpieces, then a 6-dimensional vector will be generated randomly as (2.40, 1.71, -0.89, 3.10, -2.34, -1.20). The workpiece corresponding to the smallest element (-2.34) should be processed earlier than the other workpieces. Similarly, the processing sequence can be derived as (5 6 3 2 1 4).

The search ability of the improved PSO directly depends on the inertia weight W. Since the value of W decreases linearly, the algorithm does better in global search at the beginning and in local search at the end. The inertia weight can be expressed as $W = W_{max} - \frac{W_{max}-W_{min}}{N_{max}} N$, with W_{max} and W_{min} being the initial and final values of the inertia weight, respectively, N_{max} being the maximum number of iterations, and N being the current number of iterations. Here, the initial and final values of the inertia weight are set to 1.2 and 0.4, respectively. In addition, the initial particles in the improved PSO were randomly generated by

$x^0 = x_{min} + (x_{max} - x_{min}) * r_1$ and $v^0 = v_{min} + (v_{max} - v_{min}) * r_2$,

$x_{min} = 0, x_{max} = 4, v_{min} = -4$ and $v_{max} = 4$.

4. Online no-wait scheduling based on improved PSO

As mentioned before, the leather workshop supply chain scheduling problem does not allow the rescheduling of the current workpieces when new ones are inserted into the processing sequence. Moreover, our problem is subjected to the no-wait constraint: a workpiece should be transferred to the next machine immediately after completing the process on the current machine, unless it is being transferred between two machines, and the transfer time is negligible. In addition, each workpiece must be processed in a specified time interval. To solve the problem, this section attempts to develop an online no-wait scheduling plan for the leather workshop supply chain based on the improved PSO.

4.1. Problem description

Let m be the number of the types of rubber used to process the leather workpieces, each of which is stored in n_i containers, where $i = 1, 2 \cdots, m$, a_{ij}^{k} and b_{ij}^{k} be the start time and the end time of the k-th allowable processing interval for the container j containing rubber i, respectively, and x_i and x_{m+1} be the start time and end time of a leather workpiece being processed by rubber i.

Then, the containers containing the same rubber were allocated into the same class. Meanwhile, the allowable processing intervals were ranked in ascending order of a_{ij}^{k}. The intervals with the same start time a_{ij}^{k} were sorted by the value of b_{ij}^{k}. In this way, a processing time sequence was obtained as $S_i = \{(\alpha_{ir_i}, \beta_{ir_i})\}$, with $r_i = 1,2 \cdots, q_i$. Note that q_i is the total number of containers containing rubber i.

Thus, the goals of the online no-wait scheduling problem can be described as finding the $x_1, x_2, \cdots, x_m, x_{m+1}$ such that the value of x_{m+1} is minimized and in line with the following inequalities:

$$\theta_i \le x_{i+1} - x_i \le \theta_i + \delta_i, i = 1,2, \cdots, m$$

$$x_i \ge \alpha_{ir_i}$$

$$x_{i+1} \ge \beta_{ir_i}$$

Based on the improved PSO, a real-time algorithm was designed for the online no-wait scheduling problem, with the aim to minimize the makespan under the given processing sequence. With the growing number of new workpieces, the processing sequence of these workpieces can be determined through iterations in the PSO module.

4.2. Real-time scheduling algorithm

The PSO-based real-time algorithm for the online no-wait scheduling of leather workshop supply chain was designed in the following steps:

(1) Configure the population size and the maximum number of iterations, considering the problem features.

(2) Assuming there are N workpieces, initialize the particle swarm through random generation of the position and velocity of each particle, and import some of the workpieces to the processing system according to the initial processing sequence.

(3) For each type of rubber, the sequence of allowable processing intervals can be described as $S_i = \{(\alpha_{ir_i}, \beta_{ir_i})\}$, $r_i = 1,2 \cdots, q_i$, $i = 1,2, \cdots, m$.

(4) For each $i = 1,2, \cdots, m$, $r_i = 1$.

(5) Construct a time sequence using the following equation:

$X = \{x_1, x_2, \cdots, x_{m+1}\}$.

$t_1 = \alpha_{ir_1}$

$t_i = \max(\alpha_{ir_i}, t_{i-1} + \theta_{i-1}) \ \ i = 1,2, \cdots, m$

$t_{m+1} = t_m + \theta_m$

$x_{m+1} = t_{m+1}$

$x_i = \max(t_i, x_{i+1} - \theta_i - \delta_i) \ \ i = m, m-1, \cdots, 1$

(6) If $x_{i+1} \leq \beta_{ir_i}$ for all i, go to Step (8); otherwise, go to Step (7).

(7) For all i satisfying $x_{i+1} > \beta_{ir_i}$, let $r_i = r_i + 1$ and return to Step (5).

(8) Update the sequence of allowable processing intervals; rearrange the available processing intervals of the container according to the proposed method, yielding the updated processing interval sequence $S_i = \{(\alpha_{ir_i}, \beta_{ir_i})\}$, $r_i = 1,2 \cdots, q_i$, $i = 1,2, \cdots, m$.

(9) According to the processing sequence of the workpieces, select the next workpiece entering the processing system and go to Step (4). After all workpieces have been processed, go to Step (10).

(10) Obtain the values of the following parameters: the makespan of each particle, the value of the objective function, the local optimal solution p_{best} and the global optimal solution g_{best}.

(11) By the improved PSO, calculate the position and velocity of each particle in the new generation swarm, and update the local and global optimal solutions.

(12) When the maximum number of iterations is satisfied, output the optimal value of the fitness function and terminate the iteration. Otherwise, return to Step (3). From Step (5) of the algorithm, the makespan x_{m+1} of the last process of each workpiece can be obtained. The time sequence $X = \{x_1, x_2, \cdots, x_m, x_{m+1}\}$ is the shortest makespan for each process.

Therefore, the shortest makespan of all workpieces can be determined by the above algorithm after determining the workpiece processing sequence.

5. Experimental verification

This section verifies the effectiveness of the proposed PSO-based real-time algorithm for the online no-wait scheduling of leather workshop supply chain through four examples.

5.1. Example 1

In this example, three leather workpieces need to be soaked in three kinds of chemical solutions. The first chemical solution are stored in four throughs, the second in three throughs and the third in three throughs. The allowable processing intervals for these throughs are as follows.

The first chemical solution:

Trough 1:$[0,4]$; $[6,9]$; $[10,17]$; $[29,+\infty)$

Trough 2:$[0,5]$; $[7,12]$; $[20,27]$; $[40,+\infty)$

Trough 3:$[3,6]$; $[8,9]$; $[15,19]$; $[30,+\infty)$

Trough 4:$[2,5]$; $[7,9]$; $[23,+\infty)$

The second chemical solution:

Trough 1:$[1,7]$; $[9,18]$; $[24,+\infty)$

Trough 2:$[3,6]$; $[9,12]$; $[20,24]$; $[30,+\infty)$

Trough 3:$[2,5]$; $[8,10]$; $[35,+\infty)$

The third chemical solution:

Trough 1:$[5,9]$; $[11,15]$; $[28,+\infty)$

Trough 2:$[3,10]$; $[12,17]$; $[32,+\infty)$

Trough 3:$[4,8]$; $[9,15]$; $[18,29]$; $[35,+\infty)$

The allowable processing intervals for each chemical solution were ranked in ascending order as below.

Chemical solution 1:

$S_1=\{[0,4]; [0,5]; [2,5]; [3,6]; [6,9]; [7,9]; [7,12]; [8,9]; [10,17]; [15,19]; [20,27]; [23,+\infty); [29,+\infty); [30,+\infty);[40,+\infty)\}$.

Chemical solution 2:

$S_2=\{[1,7]; [2,5]; [3,6]; [8,10]; [9,12]; [9,18]; [20,24]; [24,+\infty); [30,+\infty);[35,+\infty)\}$.

Chemical solution 3:

$S_3=\{[3,10]; [4,8]; [5,9]; [9,15]; [12,17]; [18,29]; [28,+\infty);[32,+\infty);[35,+\infty)\}$.

Then, the three leather workpieces should be processed through three intervals, respectively.

Workpiece 1: [4,5]; [3,4]; [4,6]

Workpiece 2: [2,4]; [3,4]; [6,7]

Workpiece 3: [3,4]; [5,7]; [2,4]

Next, computer simulations were performed on this example using the proposed real-time scheduling algorithm. The initial swarm size was set according to the size of the problem, that is, the number of particles was selected in light of the size of the workpieces. For Problem 1, the swarm size and the number of iterations were set to 10 and 32, respectively. The computer simulations show that the makespan was shortened to 32 through 10 iterations of 10 initial particles, indicating the good computing ability and convergence of the proposed algorithm in online scheduling problems. The Gantt chart of the optimal scheduling plan for the three workpieces is shown in Figure 1 below.

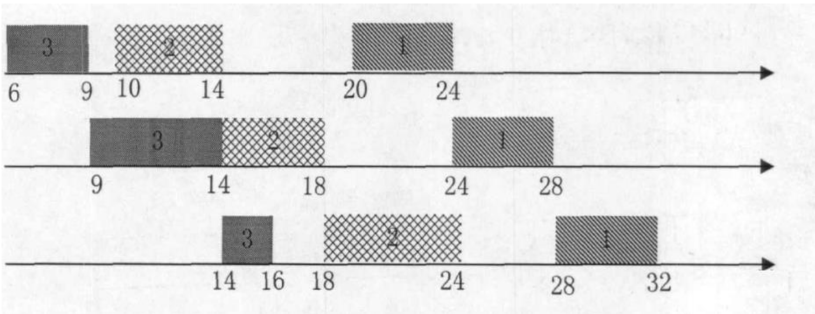

Figure 1. Optimal scheduling plan for the three workpieces

5.2. Example 2

This example was cited to verify how the proposed algrotihm schedules leather workshop supply chain on a greater scale than that in Example 2. In this example, there are 10 leather workpieces, each of which needs to be treated by 8 kinds of chemical solutions. Each chemical solution is stored in more than one trough. Then, the allowable processing intervals of each chemical solution can be expressed as:

S_1 ={[0,3]; [0,5]; [2,5]; [6,9]; [7,13]; [8,10]; [10,16]; [15,19]; [17,25]; [21,28]; [30,37]; [40, $+\infty$); [42, $+\infty$);[47, $+\infty$);[54, $+\infty$);[60, $+\infty$)}

S_2 ={[1,6]; [2,5]; [4,6]; [7,10]; [9,12]; [9,18]; [15,19]; [20,26]; [27,30]; [32,37]; [42,49]; [53, $+\infty$); [55, $+\infty$); [62, $+\infty$); [71, $+\infty$); [77, $+\infty$)}

S_3 ={[4,10]; [5,8]; [5,11]; [9,14]; [11,16]; [12,17]; [18,26]; [32, $+\infty$); [40, $+\infty$); [46, $+\infty$); [57, $+\infty$)}

S_4 ={[3,6]; [4,7]; [7,11]; [8,12]; [9,18]; [15,19]; [20,26]; [29,36]; [32,39]; [42,49]; [55, +∞); [60, +∞);[65, +∞); [71, +∞); [74, +∞)}

S_5 ={[2,4]; [4,8]; [4,9]; [7,14]; [9,12]; [9,16]; [13,19]; [18,26]; [21,32]; [31,35]; [40,47]; [49, +∞);[50, +∞); [53, +∞); [57, +∞); [60, +∞)}

S_6 ={[7,9]; [[7,13]; [8,12]; [9,15]; [13,19]; [15,19]; [21,27]; [24,30]; [30,37]; [32,40]; [43,67]; [53, +∞); [55, +∞);[61, +∞);[64, +∞);[69, +∞)}

S_7 ={[9,16]; [13,18]; [15,18]; [20,27]; [28,35]; [35,39]; [42,49]; [48,57]; [59,65]; [70,87]; [72, +∞); [77, +∞); [80, +∞); [88, +∞)}

S_8 ={[10,16]; [12,15]; [14,18]; [17,22]; [19,28]; [19,29]; [24,31]; [28,39]; [37,50]; [48,57]; [53,60]; [68,76]; [74,90]; [78, +∞);[80, +∞);[82, +∞);[86, +∞);[91, +∞)}

The allowable processing intervals of each workpiece are listed in Table 1 below.

Table 1. The allowable processing intervals of each workpiece

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
J_1	[2, 3]	[3, 5]	[3, 4]	[2, 4]	[4, 6]	[3, 5]	[2, 3]	[3, 4]
J_2	[1, 2]	[3, 4]	[2, 4]	[1, 3]	[3, 5]	[4, 5]	[3, 4]	[2, 4]
J_3	[3, 4]	[3, 5]	[2, 4]	[1, 3]	[3, 5]	[2, 3]	[2, 4]	[1, 2]
J_4	[4, 5]	[2, 3]	[1, 2]	[1, 3]	[2, 4]	[3, 4]	[2, 4]	[1, 2]
J_5	[3, 5]	[2, 3]	[1, 3]	[2, 3]	[3, 4]	[2, 4]	[2, 4]	[3, 4]
J_6	[2, 3]	[3, 4]	[3, 4]	[2, 3]	[5, 6]	[2, 3]	[2, 3]	[4, 4]
J_7	[3, 5]	[3, 4]	[2, 3]	[1, 2]	[2, 4]	[2, 3]	[1, 3]	[1, 2]
J_8	[2, 4]	[3, 5]	[2, 3]	[2, 3]	[3, 4]	[2, 4]	[2, 3]	[1, 2]
J_9	[3, 4]	[2, 3]	[1, 3]	[2, 4]	[3, 4]	[2, 3]	[4, 5]	[2, 3]
J_{10}	[3, 5]	[2, 4]	[1, 2]	[3, 4]	[2, 3]	[3, 5]	[4, 5]	[2, 3]

The swarm size and the number of iterations were set to 20 and 10, respectively. The optimal makepsan was 76 through computer simulations using the proposed real-time scheduling algorithm. This means the proposed algorithm still shows good computing ability and convergence despite the growth in the scale of the problem.

5.3. Example 3

This example was cited to verify how the proposed algorithm schedules leather workshop supply chain under different process requirements. In this example, each of the four workpieces need to be soaked in three kinds of chemical solutions. Each kind of chemical solution is stored in separate troughs. There are three troughs for the first solution, four troughs for the second solution, and three troughs for the third solution. The allowable processing intervals for these troughs are as follows.

The first chemical solution:

Trough 1: [0,6]; [8,16]; [18, $+\infty$)

Trough 2: [3,7]; [9,16]; [17,27]; [30, $+\infty$)

Trough 3: [2,8]; [9,14]; [16,20]; [25, $+\infty$)

The second chemical solution:

Trough 1: [1,7]; [9,17]; [21, $+\infty$)

Trough 2: [0,6]; [8,16]; [18,27]; [30, $+\infty$)

Trough 3: [1,5]; [7,13]; [25, $+\infty$)

Trough 4: [5,9]; [15, $+\infty$)

The third chemical solution:

Trough 1: [4,12]; [14,19]; [20, $+\infty$)

Trough 2: [5,9]; [12,27]; [35, $+\infty$)

Trough 3: [3,8]; [9,15]; [17,29]; [30, $+\infty$)

For each chemical solution, the allowable processing intervals were sorted in ascending order as below.

The first chemical solution:

$S_1 = \{$[0,6]; [2,8]; [3,7]; [8,16]; [9,14]; [9,16]; [16,20]; [17,27]; [18, $+\infty$); [25, $+\infty$); [30, $+\infty$)$\}$

The second chemical solution:

$S_2 = \{$[0,6]; [1,5]; [1,7]; [5,9]; [7,13]; [8,16]; [9,17]; [15, $+\infty$);[18,27]; [21, $+\infty$); [25, $+\infty$); [30, $+\infty$)$\}$

The third chemical solution:

$S_3 = \{$[3,8]; [4,12]; [5,9]; [9,15]; [12,27]; [14,19]; [17,29]; [20, $+\infty$); [30, $+\infty$); [35, $+\infty$)$\}$

The swarm size and the number of iterations were both set to 10. The optimal makepsan was 29 through computer simulations using the proposed real-time scheduling algorithm, revealing that each particle can reach the optimal solution after 10 iterations. The optimal scheduling sequence of the workpieces was determined as (3 4 1 2). In addition, workpiece 1 should be soaked in the three throughs in intervals of [5, 9], [9,14] and [14,17], workpiece 2 in [10,17], [17,23] and [23,27], workpiece 3 in [3, 8], [8, 16] and [16, 23] and workpiece 4 in [9, 15], [15,21] and [21,29].

5.4. Example 4

This example was cited to verify how the proposed algorithm schedules leather workshop supply chains of different scales. In these supply chains, the number of leather workpieces is 2, 3, $\cdots$, 10, respectively. Each workpiece needs to be soaked in 6 kinds of chemical solutions. The allowable processing intervals of each kind of chemical solution are given below.

$S_1 = \{[0,6]; [2,8]; [4,10]; [7,13]; [8,15]; [10,20]; [13,19]; [15,25]; [21,39]; [25,37]; [39, +\infty); [42, +\infty);[50, +\infty);[53, +\infty);[59, +\infty)\}$

$S_2 = \{[2,8]; [3,9]; [5,13]; [11,20]; [12,19]; [15,25]; [20,26]; [27,40]; [32,47]; [43, +\infty); [46, +\infty); [48, +\infty); [52, +\infty); [60, +\infty)\}$

$S_3 = \{[3,10]; [5,9]; [6,13]; [7,19]; [10,18]; [12,27]; [18,29]; [30, +\infty); [38, +\infty); [47, +\infty); [49, +\infty); [55, +\infty)\}$

$S_4 = \{[3,8]; [5,12]; [6,17]; [7,16]; [10,18]; [12,19]; [20,36]; [29,40]; [32,39]; [42,49]; [43, +\infty); [49, +\infty);[55, +\infty); [60, +\infty); [61, +\infty)\}$

$S_5 = \{[2,8]; [5,14]; [7,24]; [9,22]; [10,16]; [13,29]; [15,26]; [20,39]; [30,47]; [38, +\infty);[43, +\infty); [49, +\infty); [52, +\infty); [55, +\infty)\}$

$S_6 = \{[7,19]; [8,15]; [9,18]; [13,19]; [21,28]; [24,35]; [30,37]; [32,40]; [43,57]; [47, +\infty); [53, +\infty);[59, +\infty);[65, +\infty);[78, +\infty)\}$

The allowable processing intervals for each workpiece are listed in Table 2 below.

Table 2. The allowable processing intervals for each workpiece

	O_1	O_2	O_3	O_4	O_5	O_6
J_1	[3, 4]	[5, 7]	[3, 5]	[6, 8]	[7, 8]	[4, 5]
J_2	[2, 3]	[4, 5]	[4, 6]	[6, 7]	[4, 5]	[4, 6]
J_3	[3, 4]	[4, 5]	[6, 7]	[5, 7]	[3, 5]	[5, 6]
J_4	[4, 5]	[3, 4]	[6, 8]	[7, 8]	[5, 6]	[3, 5]
J_5	[4, 5]	[3, 4]	[5, 6]	[3, 4]	[4, 5]	[6, 7]
J_6	[5, 6]	[4, 5]	[3, 5]	[3, 4]	[4, 6]	[3, 4]
J_7	[4, 5]	[3, 5]	[5, 6]	[5, 7]	[7, 8]	[4, 5]
J_8	[4, 5]	[3, 4]	[4, 6]	[3, 4]	[4, 5]	[5, 7]
J_9	[6, 8]	[4, 5]	[3, 4]	[5, 7]	[4, 5]	[6, 8]
J_{10}	[5, 6]	[7, 9]	[6, 8]	[4, 5]	[8, 9]	[3, 5]

The nine scheduling problems were simulated on the same computer with the swarm size of 40 and the number of iterations of 10. The minimum makespan for each of the problem is recorded in Table 3.

Table 3. The shortest makespan of the 9 scheduling problems

Scale	2 × 6	3 × 6	4 × 6	5 × 6	6 × 6	7 × 6	8 × 6	9 × 6	10 × 6
Optimal solution	35	35	35	39	39	45	45	56	58

The scheduling results of one of the 9 problems are detailed below. For this problem, the swarm size and the number of iterations were set to 20 and 10, respectively. After ten simulations, the minimium makespan was obtained as 45, and the optimal processing sequence is (6 4 2 8 7 3 1 5). The initial immersion time of each workpiece in each trough can be written as the following matrix:

$$A = \begin{Bmatrix} 3 & 8 & 12 & 15 & 18 & 22 & 25 \\ 3 & 8 & 12 & 20 & 27 & 32 & 35 \\ 2 & 4 & 10 & 12 & 18 & 22 & 26 \\ 11 & 16 & 20 & 24 & 27 & 31 & 36 \\ 15 & 20 & 24 & 29 & 34 & 41 & 45 \\ 5 & 8 & 13 & 20 & 26 & 31 & 36 \\ 8 & 12 & 19 & 24 & 32 & 39 & 43 \\ 8 & 13 & 17 & 23 & 27 & 31 & 37 \end{Bmatrix}$$

The last column of the matrix is the makespan of each workpiece. The first six elements in the first row (3 8 12 15 18 22) means the start time of workpiece 6 in troughs 3, 6, 1, 4, 2 and 5, respectively, while the last element 25 means the end time of the last process of workpiece 6. Similary, the first six elements in the second row (3 8 12 15 20 27 32) means the start time of workpiece 4 in troughs 1, 6, 3, 4, 5 and 2, respectively, while the last element 35 means the end time of the last process of workpiece 4. The meanings of the elements in the other rows can be determined in analogy according to the optimal processing sequence.

6. Conclusions

After introducing the online scheduling problem of no-wait leather workshop supply chain, this paper improves the PSO algorithm, a desirable tool for continuous scheduling problems, and applies it to schedule several leather workshop supply chains. The application results show that our algorithm can minimize the makepsan of each workpiece and determine the optimal processing sequence with a small swarm and through a limited number of iterations, despite the huge amount of orders. The research findings shed new light on the management of actual leather workshop supply chains.

Acknowledgements

The authors acknowledge funding from the Chongqing Social Science Planning Project (Project No. 2018YBJY103) and the contributions from all partners of the mentioned projects.

References

Cagnina L., Esquivel S., Gallard R. (2004). Particle swarm optimization for sequencing problems: A case study. *Congress on Evolutionary Computation*, pp. 536-540. https://doi.org/10.1109/CEC.2004.1330903

Eberhart R. C., Kennedy J. (1995). A new optimizer using particle swarm theory. *International Symposium on Micromachine and Human Science,* pp. 39-43. https://doi.org/10.1109/MHS.1995.494215

Gu F., Chen S. P., Lu B. Y., Gu C. S. (2005). Particle swarm optimization for flexible job shop scheduling. *System Engineering*, Vol. 23, No. 9, pp. 20-23. https://doi.org/10.3969/j.issn.1001-4098.2005.09.006

Kennedy J., Eberhart R. C. (1995). Particle swarm optimization. *Proc. IEEE Intl. Conf. on Neural Networks, IEEE Service Center, NJ*, Vol. 4, No. 8, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Liao C. J., Tseng C. T., Luarn P. (2007). A discrete version of particle swarm optimization for fiowshop scheduling problems. *Computers & Operations Research*, Vol. 34, No. 1, pp. 3099-3111. https://doi.org/10.1016/j.cor.2005.11.017

Pan Q. K., Tasgetiren M. F., Liang Y. C. (2005). A discrete panicle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion. *International Workshop on UK Planning and Scheduling Special Interest Group*, pp. 31-41.

Pan Q. K., Wang W. H., Zhu J. Y. (2007). A discrete particle swarm optimization algorithm for solving no-wait flow shop scheduling problem. *Computer Integrated Manufacturing System*, Vol. 13, No. 6, pp. 1127-1130. https://doi.org/10.1016/j.cor.2006.12.030

Proth J. M. (2007). Scheduling: New trends in industrial environment. *Annual Reviews in Control*, Vol. 31, No. 11, pp. 157-166. https://doi.org/10.1016/j.arcontrol.2007.03.005

Schuster C. J., Framinan J. M. (2003). Approximative procedures for no-wait job shop scheduling. *Operations Research Letters,* Vol. 31, No. 7, pp. 308-318. https://doi.org/10.1016/S0167-6377(03)00005-1

Tasgetiren M. F., Liang Y. C., Sevkli M. (2004). Particle swarm optimization algorithm for makespan and maximum lateness minimization in perrnutation flowshop sequencing problem. *International Symposium on Intelligent Manufacturing Systems*, pp. 431-441. https://doi.org/10.1109/CEC.2004.1331062

Tasgetiren M. F., Liang Y. C., Sevkli M. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in permutation flowshop sequencing problem. *European Journal of Operational Research*, Vol. 177, No. 6, pp. 1930-1947. https://doi.org/10.1016/j.ejor.2005.12.024

Tasgetiren M. F., Sevkli M., Liang Y. C. (2004). Particle swarm optimization algorithm for single machine total weighted tardiness problem. *IEEE Congress on Evolutionary Computation*, pp. 1412-1419. https://doi.org/10.1109/CEC.2004.1331062

Xia W. J., Wu Z. M., Zhang W. (2004). A New hybrid optimimtion algorithm for the job-shop scheduling problem. *American Control Conference*, pp. 5552-5557.

Zheng Y., Wu B. (2011). The subject-based simulation and cluster intelligence studied by birds and ants. *Microcomputer World*, Vol. 8, No. 1, pp. 7-13.