An improved simulated annealing algorithm based on residual network for permutation flow shop scheduling

Yang Li¹ · Cuiyu Wang¹ · Liang Gao¹ · Yiguo Song² · Xinyu Li¹

Received: 30 June 2020 / Accepted: 18 September 2020 / Published online: 8 October 2020
© The Author(s) 2020

Abstract

The permutation flow shop scheduling problem (PFSP), which is one of the most important scheduling types, is widespread in the modern industries. With the increase of scheduling scale, the difficulty and computation time of solving the problem will increase exponentially. Adding the knowledge to intelligent algorithms is a good way to solve the complex and difficult scheduling problems in reasonable time. To deal with the complex PFSPs, this paper proposes an improved simulated annealing (SA) algorithm based on residual network (SARes). First, this paper defines the neighborhood of the PFSP and divides its key blocks. Second, the Residual Network (ResNet) is used to extract and train the features of key blocks. And, the trained parameters are stored in the SA algorithm to improve its performance. Afterwards, some key operators, including the initial temperature setting and temperature attenuation function of SA algorithm, are also modified. After every new solution is generated, the parameters trained by the ResNet are used for fast ergodic search until the local optimal solution found in the current neighborhood. Finally, the most famous benchmarks including part of TA benchmark are selected to verify the performance of the proposed SARes algorithm, and the comparisons with the state-of-art methods are also conducted. The experimental results show that the proposed method has achieved good results by comparing with other algorithms. This paper also conducts experiments on network structure design, algorithm parameter selection, CPU time and other problems, and verifies the advantages of SARes algorithm from the aspects of stability and efficiency.

Keywords Permutation flow shop scheduling · Improved simulated annealing algorithm · Residual networks · TA benchmark

Introduction

Scheduling is an indispensable part of the modern manufacturing process. Intelligent workshop scheduling can not only ensure the orderly progress of workshop manufacturing process but also maximize the utilization of resources and reduce the waste in the manufacturing process, thus reducing the production and manufacturing cost [28]. The permutation flow shop scheduling problem (PFSP), which is one of the most important scheduling types, is widespread in the modern industries, including automobile [35], electronic [5], chemical [20] and other industries. Therefore, the effective

PFSP algorithm can improve the productivity of these industries well.

However, the PFSP is a well-known NP—hard problem and is very hard to be solved in large scales. At present, most of the PFSP methods mainly focus on the meta-heuristic algorithms, such as genetic algorithm (GA) [30], simulated annealing algorithm (SA) [10], tabu search algorithm (TS) [11], particle swarm optimization algorithm (PSO) [39], etc. In recent years, there have been some related research works. Various meta-heuristic algorithms emerge one after another, which provides a lot of theoretical basis for the study of PFSP [29, 31, 32]. Suresh et al. [25] proposed a social group optimization (SGO), which is a population-based optimization technique. Taking advantage of both epsilon greedy and Levy flight, Liu et al. [16] proposed a greedy–Levy Ant colony optimization (ACO) incorporating these two approaches to solve the complicated combinatorial optimization problems. Sayed et al. [22] presented a hybrid algorithm based on moth-flame optimization (MFO) algorithm with simulated annealing (SA), namely (SA-MFO). It can escape from local

✉ Xinyu Li
lixinyu@mail.hust.edu.cn

¹ The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, China

² The State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China

optima mechanism of SA and fast searching and learning mechanism for guiding the generation of candidate solutions of MFO.

Although these meta-heuristic algorithms have some advantages, the solution quality may be poor and the computation time may be long when dealing with the large-scale and complex PFSPs [7]. Because, the previous historical data cannot be used to mine the variation rules of scheduling, especially for large-scale and complex problems, which may fail to achieve good results, and the calculation time will increase exponentially [13]. Therefore, to enhance the optimization capability when solving complex problems, it is very important to incorporate the knowledge into the algorithm. Adding the knowledge to intelligent algorithms is a good way to solve the complex and difficult scheduling problems in reasonable time, including the PFSPs [18]. Therefore, this paper adopts a knowledge-driven method to assist the intelligent algorithm for the PFSPs.

But, how to extract the knowledge from the historical data is a very challenging work. Deep learning is a very effective method to extract knowledge based on historical data and some prior knowledge to form its own knowledge system and learning skills. Yoshua et al. [37] surveys the recent attempts, both from the machine learning and operations research communities, at leveraging machine learning to solve the combinatorial optimization problems.

Therefore, this paper uses the residual networks (ResNet), a very good and simple deep learning method, as a tool to train and classify the features for the PFSPs. By quickly judging the neighborhood of the problem, the intelligent algorithm can traverse as many solutions as possible in a very short time. Because the network training process can precede the start of scheduling, precious scheduling time is not consumed. This way can not only improve the effectiveness of the algorithm but also reduce the computation time.

As a kind of convolutional neural networks (CNN), ResNet is one of the most commonly used deep learning methods. In 2015, He et al. [8] used ResNet to solve the problem of gradient disappearance. Valeryi et al. [26] proposed the technique of automatic labeling the data set with the finite element model for training of artificial neural network in tomography. Lin et al. [15] proposed a residual networks of residual networks (RoR) optimization method to avoid over-fitting and gradient vanish. Gomez et al. [6] found in the ResNet network along with the increase of the depth in improving performance, but increased memory consumption, thus put forward each layer's activations can be reconstructed exactly from the next layer's. Li et al. [14] propose a multi-scale residual network to solve the problem of super-resolution in different scales of a single image. Zhong et al. [40] inspired by the residual network, designed an end-to-end spectral-spatial residual network (SSRN) that takes

raw 3-D cubes as input data without feature engineering for hyperspectral image classification.

The main feature of ResNet is the cross-layer connectivity, where can transfer inputs across layers and add to the results of convolution by introducing shortcut connections. Deep neural networks are very difficult to train [19]. The ResNet proposes a framework for learning residuals to simplify the training of networks that are deeper than those previously used. Instead of learning unknown functions, it has explicitly defined learning residual functions. Comprehensive empirical evidences indicate that the ResNet is easier to optimize and its accuracy can be obtained with significantly increased depth. There is only one pooling layer in ResNet, which is connected behind the last convolutional layer. ResNet enables the underlying network to be fully trained, and the accuracy rate is significantly improved with the deepening of the depth. ResNet, with a depth of 152 layers, won first place in the ISVRC-15 image classification competition. With its good performance, this paper uses it to classify the extracted workshop characteristic data. According to the classification results, it can quickly judge whether the current neighborhood search operation is effective or not.

Since the ResNet assisted method proposed in this paper can be used to improve the most meta-heuristic algorithms, to verify its effectiveness, an improved SA algorithm based on ResNet (SARes) is proposed for PFSP in this paper. The most famous benchmarks of PFSP including part of TA benchmark are selected to verify the performance of the proposed algorithm, and the comparisons with the state-of-art methods are also conducted. The experimental results show that the proposed SARes method has achieved significant results. This paper also conducts experiments on network structure design, algorithm parameter selection, CPU time and other problems, and verifies the advantages of SARes algorithm from the aspects of stability and efficiency.

The rest of this paper is organized as follows: Sect. 2 introduces the PFSP. Section 3 provides the proposed SARes for PFSP. Section 4 gives the experimental results and comparisons with the state-of-art algorithms. The final section shows the conclusion and future work.

Problem formulation

Introduction of permutation flow shop scheduling problem

In 1954, Johnson [9] first proposed and studied the PFSP. Since then, PFSP has attracted the attentions from a large number of scholars. After decades of research, a lot of good algorithms had been proposed to solve such problems [27].

In the PFSP, the solutions are represented by the permutation of n jobs, i.e., $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$. Each job contains

m operations, and every operation is performed by a different machine. Jobs, once initiated, cannot be interrupted (preempted) by another job on each machine and the release time of all jobs is zero. Thus, given the processing time t_{jk} for the job j on the machine i , the PFSP is to find the best permutation of jobs $\sigma^* = \{\sigma_1^*, \sigma_2^*, \dots, \sigma_n^*\}$ to be processed on each machine to optimize the objectives, such as the makespan. $C(\sigma_j, m)$ denotes the completion time of the job σ_j on the machine m [18].

Then given the job permutation σ , the completion time for the n -job, m -machine problem is calculated as follows:

$$\begin{cases} C_{\sigma_1,1} = t_{\sigma_1} \\ C_{\sigma_1,j} = C_{\sigma_1,j-1} + t_{ij}, & j = 2, 3, \dots, m \\ C_{\sigma_i,1} = C_{\sigma_{i-1},1} + t_{i1}, & i = 2, 3, \dots, n \\ C_{\sigma_i,j} = \max(C_{\sigma_{i-1},j}, C_{\sigma_i,j-1}) + t_{ij}, \\ i = 2, 3, \dots, n; & j = 2, 3, \dots, m \end{cases} \quad (1)$$

t_{ij} represents the processing time of job i on machine j . The best solution is with the minimum makespan.

Although the PFSP is a special type of flow shop scheduling, it is still a very complex combinatorial optimization problem [36]. With the increase of jobs, the size of the solution space will increase exponentially, so it is extremely difficult to find a satisfactory solution for the large scale PFSPs.

Definition of neighborhood for PFSP

In this study, the neighborhood search of PFSP is defined as follows: Exchange the machining order of all adjacent jobs in turn, update if the solution quality becomes better, otherwise continue to exchange the next set of jobs. Until any two adjacent jobs in the sequence are swapped and the result cannot be improved, the local optimal solution under the current neighborhood is obtained by default.

As shown in Fig. 1, this is the Gantt chart of the PFSP. Suppose we currently need to determine whether the new solution after exchanging jobs 8 and 5 has been improved. The traditional approach is to recalculate the makespan as a whole. This will take a lot of time, which will affect the final solution efficiency.

The analysis of Gantt chart shows that the order of the two sides of the red area does not change. Therefore, the results can only be judged by the changes in the red area. For the example given in the figure, the result of swapping 8 and 5 is only related to the end of 9, the beginning of 8 and 5, the end of 8 and 5, and the beginning of 11. In this paper, the red areas are defined as key blocks.

If we can find the relationship, we can quickly determine the quality of the solution and reduce the computation time.

However, it is difficult to find relevant laws artificially, so artificial intelligence is needed.

Proposed SAREs for PFSP

According to the analysis of the current situation of the PFSP problem, the combination explosion problem must be dealt with before the problem to be solved. When the scale of the problem grows, the ergodic solution takes too long to get the result. However, algorithms with stochastic properties such as meta-heuristic may fall into the local optimal solutions. Therefore, the knowledge-driven approach is adopted to speed up the search process as much as possible, so that the meta-heuristic can search more neighborhoods within a limited time.

Introduction of ResNet

The basic structure of CNN is composed of the input layer, the convolutional layer, the pooling layer (also known as the sampling layer), the full connection layer and the output layer. It has the structural characteristics of local area connection, weight sharing and down sampling. Its development has gone through several critical periods:

In 2015, He et al. [8] used residual networks (ResNet) to solve the problem of gradient disappearance. The main feature of ResNet is cross-layer connectivity, where you transfer inputs across layers and add to the results of convolution by introducing shortcut connections.

Deep network has the problem of gradient disappearance or explosion, which makes deep learning model difficult to train. The degradation of the deep network at least indicates that deep network is not easy to train. So, ResNet is used to solve this problem.

The structure of the residual block is shown in Fig. 2. In general network structure, the input x_1 directly passes through two convolutional layers to obtain the output $x_1 + 1$, while the residual block is to add the output through two convolutional layers to the network input x_1 . We can call this method as skip connection. Skip Connection is not only allowed to add the input x_1 and convolution results directly. In some cases, the input x_1 and $x_1 + 1$ dimension are different, so 1×1 convolution can be added to reduce the dimension of the input so that x_1 and $x_1 + 1$ dimension are the same, and the two can be added.

Suppose the input is X , and there are two fully bonded layers that asymptotically fit $H(x)$. Assuming that $H(x)$ is the same as x dimension, then fitting $H(x)$ is equivalent to fitting residual function $H(x) - x$. If the residual function $F(x) = H(x) - x$, then the original function becomes $F(x) + x$. Then a cross-layer connection is directly added on the

Fig. 1 The Gantt chart of the PFSP

Fig. 2 Residual learning: a building block

basis of the original network, and the cross-layer connection here is to transfer X 's Identity Mapping.

The entire residual structure can be formally defined as:

$$y = F(x, \{Wi\}) + x. \quad (2)$$

Here $F(x, \{Wi\})$ refers to the fitting residual mapping, as shown in the Fig. 2, there are two full connection layers, $F = W_2\sigma(W_1x)$, where σ refers to ReLU. When F and x have the same dimension, they can be added to each element directly. But if not, we have to add a linear map to x and map it to a vector that has the same dimension as F , so the whole residual structure is:

$$y = F(x, \{Wi\}) + W_s x. \quad (3)$$

W_s is a matrix for dimensional matching.

Combination of residual network and scheduling

According to the above introduction, the ResNets is mainly used to solve the image classification. For scheduling, we can think of the key block matrix as an image. First, we solve the makespan directly and annotate the critical blocks. Then, a large number of labeled key blocks are used as training sets to train the residual network. Finally, the trained residual network parameters are saved and used to judge the influence of the processing order of adjacent jobs under the same scale on the result. If the network output is “better”, then we swap the current jobs. If the network output is “worse”, the original sequence is left unchanged and takes place the next iteration.

The training data used in this paper are self-generated. For the case of the same scale, the key blocks of the sequence are tagged by the traditional method and the tagged samples were used as training sets for ResNet.

Taking Fig. 1 as an example, the operation process is shown in Fig. 3.

Improved simulated annealing algorithm for PFSP

The SA algorithm originated from the Metropolis criterion proposed by Metropolis et al. in 1953 [1]. Many studies have shown that SA algorithm is a good optimization algorithm that can converge to the optimum solution with probability 1. In 1983, Kirkpatrick et al. successfully introduced the idea of annealing into the optimization field [10]. SA has been widely used in various optimization problems due to its strong local searching ability, easy operation and fast solving speed. In this paper, some improvements are made to the SA algorithm.

job	9	8		5		11
machine	End	Star	End	Star	End	Star
1	573	573	641	641	695	695
2	659	659	664	695	774	774
3	734	734	811	811	827	873
4	811	811	862	862	928	928
5	865	865	933	933	991	996

Fig. 3 Residual network operation flow chart

Encoding and parameter initialization

Encoding method adopts natural number coding: n jobs are numbered 1 – n successively. The generated scheduling scheme is consistent with the corresponding natural number coding. The initial solution is generated by NEH algorithm [24]. In this paper adopts the method of random generation of a set of state, to determine the difference between the two states biggest target “ $|\Delta \max|$ ”, then according to the difference value, the initial acceptance probability is used to determine the initial temperature [38]. The relevant expression is as follows:

$$T_0 = -\Delta \max / p_0. \quad (4)$$

Based on computational experience, the parameters are set as follows: $k = 2000$, $a = 0.95$, $P_0 = 0.2$. Among them, k is markov chain length, a is temperature attenuation coefficient and P_0 is the initial acceptance probability.

Multi-rule neighborhood search

The neighborhood has been defined in the Sect. 2, and it is proved that the local optimal solution can be found in the current neighborhood. Therefore, in the process of iteration, we need to disrupt the original sequence as much as possible to make it jump out of the local optimal solution. In this paper, three ways of generating new solutions are selected randomly according to a certain probability.

1. Binary exchange: randomly select two points in the coding sequence and reverse the order of all the jobs between them.
2. Three points exchange: randomly select three points in the coding sequence, and exchange two sequence positions between them.
3. Two points exchange: randomly select two points in the coding sequence and exchange the job positions of these two points.

For each new solution generated, the method given above is used to search through ergodic in the neighborhood until the local optimal solution under the current neighborhood is found. At the same time, to ensure that excellent solutions are not replaced in subsequent operations, we set a global variable E_best to hold the optimal solution.

Temperature decay function

The temperature attenuation function is an important parameter that affects the performance of simulated annealing algorithm [4]. The formula of Kepler-type decay function decline curve is as follow:

$$T = T_0 a^k (\cos(\pi / (2(1 - k/K))) + \cos(\pi / (2T_0(1 - k/K))), \quad (5)$$

a is temperature attenuation coefficient, k is the number of iterations, and K is the total cooling times. This function is executed every time the neighborhood search is completed until the temperature T reaches the minimum. After the temperature attenuation is completed, the optimal objective function value is returned and the program is finished. In each decay. If 200 consecutive iterations fail to update the current optimal solution, the search at the current temperature is ended.

Framework of the improved SAREs

The improved simulated annealing algorithm is shown in the Fig. 4. Driven by knowledge, the meta-heuristic algorithm increases the stable global searching capability on the basis of randomness. In each iteration at temperature, the optimal solution in the current search space can be obtained. At the same time, the learning and training time of the algorithm can be completed in advance without occupying the scheduling time, which greatly increases the overall computing efficiency of the algorithm.

Here is an example of how to combine ResNet and SA in PFSP. Suppose there are five machines and ten jobs in the factory. First, we numbered them 1 to 10. Then, the generated initial sequence was input into the SA for iteration. After each

Fig. 4 Flow chart of the proposed SA algorithm

neighborhood search described in Sect. 3.3.2 is performed, the resulting sequence is imported into the trained ResNet. Assume that the sequence at this time is (5, 4, 3, 6, 8, 7, 9, 2, 1). Next, the neighborhood search described in Sect. 3.2 begins. We need to judge whether the exchanging position of all adjacent jobs (such as 5 and 4, 4 and 3, 3 and 6) can make the result better. To make a judgment quickly, we took the key blocks in Sect. 2.2 as input and put them into the trained Resnet to get the final judgment result. Adjust the sequence according to the result until all adjacent jobs are exchanged to make the result better. With this result as the local optimal solution at the current SA temperature, restart the execution of the SA program. After the final execution of SA, the scheduling scheme under the current workshop can be obtained.

Experimental results and discussion

The experimental setup

The proposed algorithm is encoded in python and runs on a laptop with the following capabilities: Intel Core i7-8750H CPU @ 2.20 GHz with 16.0 GB Memory in Windows 10.

To facilitate the performance of the algorithm, the evaluation index includes makespan, the average relative percentage deviation (ARPD).

$$\text{ARPD} = \frac{(\sum_{i=1}^n C_i)/n - \text{UB}}{\text{UB}} \cdot 100\%, \quad (6)$$

n is the running times of the algorithm, C_i is the result of the i th running of the algorithm, and UB is the currently known best solution.

This paper selects the part of TA benchmark (number of jobs ≥ 100) to evaluate the performance of the proposed algorithm. TA benchmark is one of the most famous benchmarks, which contains the test data of various workshop scheduling models including PFSP [3].

Because when the number of jobs is less than 100, most algorithms can find the optimal solution. Therefore, this paper selects cases with more than 100 jobs for the comparative experiment (TA61–TA120). This paper also conducts experiments on network structure design, algorithm parameter selection, CPU time and other problems, and verifies the advantages of SRes algorithm from the aspects of stability and efficiency.

Experimental results of Taillard benchmark

Taillard [26] proposed TA benchmark in 1993, which contains 260 different workshop scheduling instances. It became one of the most widely used benchmarks in the field of scheduling. All problem determination of processing time and the known optimal solution can refer to <https://mistic.heigvd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>.

The selected state-of-the-art comparison algorithms are as follows:

- Self-guided differential evolution with neighborhood search (NS-SGDE, 2016) [23];
- Hybrid differential evolution (L-HDE, 2014) [33];
- Hybrid teaching learning-based optimization (HTLBO, 2014) [34];
- Estimation of distribution algorithm variable neighborhood search (EDA-VNS, 2017) [17];
- Opposition-based differential evolution (ODDE, 2013) [12];
- Extended artificial chromosomes genetic algorithm (eACGA, 2012) [2].

For each instance in the benchmark, run it 10 times and record the results, then calculate the ARPD according to the formula (6). The results of Taillard benchmark are shown in Table 1. Data statistics are shown in Table 2. The significance of bold is the best solution on the same scale in the table. Because some references (L-HDE, HTLBO, EDA-VNS) did not provide the specific makespan for every instance, only some algorithms (SRes, ISA, NS-SGDE, ODDE, eACGA) are compared in Table 1. Among them, “ISA” represents the

Table 1 Comparison on Taillard benchmark based on the best scheduling

Instance	UB	SARes	ISA	NS-SGDE [23]	ODDE [12]	eACGA [2]
TA61	5493	5493	5493	5493	5493	5493
TA62	5268	5283	5284	5283	5284	5284
TA63	5175	5175	5175	5182	5185	5177
TA64	5014	5018	5023	5018	5018	5018
TA65	5250	5250	5250	5253	5253	5253
TA66	5135	5135	5135	5135	5135	5135
TA67	5246	5246	5246	5246	5246	5246
TA68	5094	5094	5112	5094	5095	5098
TA69	5448	5448	5448	5448	5454	5454
TA70	5322	5322	5346	5322	5322	5328
TA71	5770	5780	5780	5784	5800	5783
TA72	5349	5362	5403	5362	5378	5367
TA73	5676	5691	5693	5691	5698	5679
TA74	5781	5826	5830	5826	5890	5848
TA75	5467	5491	5491	5503	5533	5512
TA76	5303	5308	5317	5308	5309	5317
TA77	5595	5610	5628	5610	5633	5618
TA78	5617	5630	5687	5630	5695	5660
TA79	5871	5882	5940	5882	5940	5926
TA80	5845	5881	5903	5881	5896	5881
TA81	6202	6310	6339	6360	6458	6407
TA82	6183	6268	6317	6278	6364	6349
TA83	6271	6384	6391	6405	6493	6421
TA84	6269	6349	6364	6394	6461	6410
TA85	6314	6421	6439	6452	6535	6475
TA86	6364	6455	6487	6461	6603	6511
TA87	6268	6372	6382	6385	6463	6441
TA88	6401	6496	6549	6496	6641	6580
TA89	6275	6362	6384	6428	6501	6458
TA90	6434	6562	6575	6538	6616	6575
TA91	10,862	10,887	10,914	10,887	10,933	10,928
TA92	10,480	10,517	10,531	10,555	10,570	10,573
TA93	10,922	10,962	11,000	10,980	11,006	11,017
TA94	10,899	10,910	10,939	10,917	10,939	10,939
TA95	10,524	10,537	10,537	10,537	10,575	10,575
TA96	10,326	10,338	10,388	10,357	10,409	10,395
TA97	10,854	10,931	10,945	10,929	10,966	10,945
TA98	10,730	10,785	10,810	10,798	10,798	10,828
TA99	10,438	10,467	10,495	10,465	10,505	10,473
TA100	10,675	10,727	10,727	10,727	10,758	10,744
TA101	11,195	11,315	11,403	11,468	11,527	11,481
TA102	11,203	11,412	11,432	11,487	11,646	11,634
TA103	11,281	11,489	11,503	11,549	11,721	11,662
TA104	11,275	11,502	11,516	11,553	11,663	11,639
TA105	11,259	11,459	11,459	11,438	11,578	11,533
TA106	11,176	11,396	11,413	11,445	11,587	11,544
TA107	11,360	11,555	11,599	11,596	11,725	11,757

Table 1 continued

Instance	UB	SARes	ISA	NS-SGDE [23]	ODDE [12]	eACGA [2]
TA108	11,334	11,561	11,581	11,592	11,699	11,692
TA109	11,192	11,393	11,471	11,485	11,556	11,540
TA110	11,288	11,487	11,533	11,607	11,679	11,660
TA111	26,059	26,374	26,442	26,420	26,639	26,763
TA112	26,520	26,951	27,029	26,942	27,167	27,214
TA113	26,371	26,727	26,810	26,729	26,962	26,978
TA114	26,456	26,742	26,790	26,751	27,027	27,022
TA115	26,334	26,579	26,690	26,643	26,787	26,928
TA116	26,477	26,814	26,865	26,832	26,939	27,127
TA117	26,389	26,574	26,790	26,609	26,754	26,931
TA118	26,560	26,925	27,010	26,925	27,107	27,173
TA119	26,005	26,311	26,372	26,326	26,477	26,535
TA120	26,457	26,766	26,826	26,766	26,954	26,924
Number of optimal solutions		54	10	26	5	5

Table 2 Comparison on Taillard benchmark based on the ARPD

Problem	SARes	ISA	NS-SGDE [23]	L-HDE [32]	HTLBO [33]	EDA-VNS [17]	ODDE [12]	eACGA [2]
100*5	0.036	0.129	0.056	0.409	0.262	0.631	0.076	0.078
100*10	0.330	0.704	0.358	1.306	1.218	1.216	0.877	0.559
100*20	1.584	1.978	1.933	4.773	3.594	1.827	3.420	2.616
200*10	0.328	0.460	0.414	1.315	1.222	0.457	0.702	0.662
200*20	1.782	2.032	2.360	2.972	2.880	1.815	3.392	3.179
500*20	1.189	1.515	1.257	1.965	1.903	1.333	1.967	2.264
ARPD	0.875	1.136	1.063	2.123	1.847	1.213	1.734	1.560

improved SA algorithm proposed in Sect. 3, which does not include the neighborhood search part. “SARes” represents a new knowledge-driven algorithm based on improved SA.

In the 60 benchmarks shown in Table 1, the knowledge-driven algorithm proposed in this paper achieves 54 optimal solutions, which is far better than other algorithms. Meanwhile, from the average value shown in Table 2, it has the lowest value with 0.875 and outperforms all other algorithms including ISA. It is also evident that the solution space clipping method is effective and reliable when solving PFSPs.

The above test results of TA benchmark show that when solving complex system problems, it is necessary to add knowledge-driven methods to improve the original intelligent algorithm.

It can be seen from the above experiments that the knowledge-driven residual network scheduling model proposed in this paper is effective and the state-of-art results in TA benchmark. The experimental results show that the knowledge-driven algorithm can solve complex system problems better than the original model.

At the same time, SARes is also a randomized method with certain instability. Therefore, this paper takes TA81 as an

example to compare the stability of SARes and ISA. The two methods are used to calculate TA81 for 10 times, respectively, and the resulting figure is shown in Fig. 5. It can be seen from the experiment that the variance of SARes and ISA is 80.27 and 1138.06, respectively. It can be seen from the comparison of Fig. 5 and variance that the SARes method proposed in this paper has a better stability.

Calculation time comparison

Since the training process of a neural network can be completed in advance and does not occupy the scheduling time, it does not affect the overall efficiency. When the scheduling algorithm is executed, only the corresponding parameters need to be invoked. Table 3 shows the average CPU time comparison of some benchmarks under the two algorithms.

In the benchmark where the number of work units is equal to 500 (TA111–TA120), the calculation time of SARes is about near to 349.5 s. However, the ISA without knowledge-driven requires close to 934.1 s, which indicates that the knowledge-driven algorithm proposed in this paper is also

Fig. 5 Algorithm stability analysis **Table 3** Comparison on Taillard benchmark based on the average CPU Time

Problem	100*5	100*10	100*20	200*10	200*20	500*20
SARes (s)	20.5	52.7	113.5	126.8	197.3	349.5
ISA (s)	70.1	94.4	180.3	231.7	455.2	934.1

conducive to improving the computational efficiency of the algorithm.

SARes has more knowledge-driven neighborhood search modules than ISA, and its algorithm complexity is bound to increase. But because of this improvement, the algorithm can converge to the local optimal solution more quickly. Both the ISA and SARes algorithms presented in this paper take the unimproved optimal solution for 200 consecutive iterations as the termination condition. Therefore, SARes has a faster computational speed than ISA and it can rapidly converge to the local optimal solution.

ResNet network parameter selection

There are four commonly used ResNet structures, namely ResNet18, ResNet50, ResNet101 and ResNet152 [8]. These four structures all have their corresponding use environments. Therefore, to determine the network structure parameters in this paper, some benchmarks are selected for experiments and compare their fitting accuracy (Table 4).

Because the data of key blocks in this paper is relatively simple, when the network structure is too complex, there will be an obvious overfitting phenomenon. Combined with the above experiments, it can be known that the most suitable network structure for this paper is ResNet50.

Meta-heuristic algorithm analysis

To prove that the proposed knowledge-driven neighborhood search algorithm is applicable to any meta-heuristic algorithm, several representative algorithms are selected for

Table 4 Comparison on Taillard benchmark based on the fitting accuracy

Problem	ResNet18	ResNet50	ResNet101	ResNet152
100*5	0.931	0.924	0.865	0.843
100*10	0.914	0.927	0.885	0.897
100*20	0.906	0.936	0.905	0.911
200*10	0.912	0.925	0.891	0.907
200*20	0.904	0.933	0.904	0.913
500*20	0.908	0.939	0.899	0.917
ARPD	0.913	0.931	0.892	0.898

comparison, including GA, PSO and IG [21]. Select the TA (TA61–TA120) benchmark as the test case, experimental results are shown in Table 5. Compared with SARes, GARes, PSORes and IGRes in terms of quality, the three algorithms (ISA, GA, PSO, IGRes) are all improved respectively, which proves that the proposed knowledge-driven neighborhood method is effective and can improve the performance of the meta-heuristic algorithm.

Conclusions and future work

This paper proposes an improved SA algorithm based on the residual network to solve the PFSP problem. This method fuses the knowledge-driven and achieves the state-of-art results in TA benchmark. At the same time, because the training process of neural network can be completed in advance, it doesn't take up valuable scheduling time.

The main contribution of this paper is to propose the knowledge-driven method to solve the PFSP. According to the experiments, to enhance the optimization capability when solving particular problems, it is very important to incorporate knowledge in intelligent algorithms.

Although the method proposed in this paper has achieved good results, there are still some limitations. First of all, this paper does not prove whether the residual neural network is

Table 5 Meta-heuristic algorithm analysis

Problem	Relative percentage deviation							
	ISA	SARes	GA	GARes	PSO	PSORes	IG	IGRes
100*5	0.129	0.036	0.082	0.071	0.094	0.080	0.051	0.046
100*10	0.704	0.330	0.976	0.842	1.052	0.892	0.806	0.744
100*20	1.978	1.584	2.823	2.011	2.658	2.455	2.347	2.291
200*10	0.460	0.328	0.914	0.648	0.874	0.796	0.748	0.693
200*20	2.032	1.782	3.583	2.799	3.479	3.117	2.764	2.731
500*20	1.515	1.189	2.892	2.314	3.017	2.697	2.541	2.533
ARPD	1.136	0.875	1.878	1.448	1.862	1.673	1.543	1.506

the most suitable network for this problem. In the future, we can consider adopting other more advanced network structures. Secondly, whenever there is a change in the number of machines in the workshop, we have to retrain the network, which takes a lot of effort. Therefore, in future work, it is necessary to further analyze the relevant network model, improve the theoretical framework, and strive to extend it to all kinds of complex system problems.

Acknowledgements This work was supported by the National Key R&D Program of China under Grant no. 2019YFB1704600, the National Natural Science Foundation of China under Grant no. 51775216 and 51825502, and in part by the Natural Science Foundation of Hubei Province under Grant no. 2018CFA078.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chen DJ, Lee CY, Hand PC, Mendes P (2007) Parallelizing simulated annealing algorithms based on high-performance computer. *J Glob Optim* 39(2):261–289
- Chen SH, Chang PC, Cheng TCE, Zhang Q (2012) A self-guided genetic algorithm for permutation flowshop scheduling problems. *Comput Oper Res* 39(7):1450–1457
- Davendra D, Bialic-Davendra M (2013) Scheduling flow shops with blocking using a discrete self-organising migrating algorithm. *Int J Prod Res* 51(8):2200–2218
- Ding JY, Song S, Gupta JND, Zhang R, Chiong R, Wu C (2015) An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Appl Soft Comput* 30:604–613
- Gao K, Huang Y, Sadollah A, Wang L (2020) A review of energy-efficient scheduling in intelligent production systems. *Comput Intell Syst* 6:237–249
- Gomez AN, Ren M, Urtasun R, Grosse RB (2017) The reversible residual network: backpropagation without storing activations. *Adv Neural Inf Process Syst* 30:2214–2224
- Gui L, Gao L, Li XY (2020) Anomalies in special permutation flow shop scheduling problems. *Chin J Mech Eng-En*. <https://doi.org/10.1186/S10033-020-00462-2>
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
- Johnson SM (1954) Optimal two- and three-stage production schedules with setup times included. *Nav Res Logist Q* 1(1):61–68
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Li X, Gao L (2016) An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *Int J Prod Econ* 174:93–110
- Li X, Yin M (2013) An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure. *Adv Eng Softw* 55:10–31
- Li XY, Lu C, Gao L, Xiao SQ, Wen L (2018) An effective multi-objective algorithm for energy efficient scheduling in a real-life welding shop. *IEEE Trans Ind Inform* 14(12):5400–5409
- Li JC, Fang FM, Mei KF, Zhang GX (2018) Multi-scale residual network for image super-resolution. In: *Proceedings of the European conference on computer vision*, pp 517–532
- Lin L, Yuan H, Guo LR (2018) Optimization method of residual networks of residual networks for image classification. *Int Conf Int Comput* 10956:212–222
- Liu YH, Cao BY, Li HH (2020) Improving ant colony optimization algorithm with epsilon greedy and Levy flight. *Comput Intell Syst*. <https://doi.org/10.1007/s40747-020-00138-3>
- Liu Y, Yin M, Gu W (2014) An effective differential evolution algorithm for permutation flow shop scheduling problem. *Appl Math Comput* 248:143–159
- Liu ZF, Yan J, Cheng Q, Yang CB, Sun SW, Xue D (2020) The mixed production mode considering continuous and intermittent processing for an energy-efficient hybrid flow shop scheduling. *J Clean Prod* 119071:1–17
- Mhapsekar M, Mhapsekar P, Mhatre A, Sawant V (2020) Implementation of residual network (ResNet) for devanagari handwritten character recognition. In: Vasudevan H, Michalas A, Shekhar N, Narvekar M (eds) *Advanced computing technologies and applications. Algorithms for intelligent systems*. Springer, Singapore, pp 137–148. https://doi.org/10.1007/978-981-15-3242-9_14
- Rahman HF, Sarker R, Essam D (2017) A real-time order acceptance and scheduling approach for permutation flow shop problems. *Oper Res* 57(4):345–347

21. Ruiz R, Stützle T (2007) A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *Eur J Oper Res* 177(3):2033–2049
22. Sayed GI, Hassanien AE (2018) A hybrid SA-MFO algorithm for function optimization and engineering design problems. *Comput Intell Syst* 4:195–212
23. Shao W, Pi D (2016) A self-guided differential evolution with neighborhood search for permutation flow shop scheduling. *Expert Syst Appl* 51:161–176
24. Sun BQ, Wang L (2020) An estimation of distribution algorithm with branch-and-bound based knowledge for robotic assembly line balancing. *Comput Intell Syst*. <https://doi.org/10.1007/s40747-020-00166-z>
25. Suresh S, Anima N (2016) Social group optimization (SGO): a new population evolutionary optimization technique. *Comput Intell Syst* 2:173–203
26. Valeryi MB, Stanislav AK, Liudmyla SK (2020) Galois field augmentation model for training of artificial neural network in dentistry data-Ce. *Bus Appl* 48:339–369
27. Wang GC, Gao L, Li XY, Li PG, Tasgetiren MF (2020) Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm. *Swarm Evol Comput* 57:100716
28. Wang JJ, Wang L (2020) A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop. *IEEE Trans Syst Man Cybern Syst* 50(5):1805–1819
29. Wang L, Wang S, Zheng X (2016) A hybrid estimation of distribution algorithm for unrelated parallel machine scheduling with sequence-dependent setup times. *IEEE/CAA J Autom Sinica* 3(3):235–246
30. Wang SY, Wang L (2016) An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem. *IEEE Trans Syst Man Cybern Syst* 46(1):139–149
31. Wang YH, Chen WY (2019) A decomposition-based hybrid estimation of distribution algorithm for practical mean-cvar portfolio optimization. In: *Proceedings of the international conference on intelligent computing*, Nanchang, China, pp 38–50
32. Wu C, Wang L (2018) A multi-model estimation of distribution algorithm for energy efficient scheduling under cloud computing system. *J Parallel Distr Com* 117:63–72
33. Rafajłowicz W (2015) A hybrid differential evolution-gradient optimization method. In: *Proceedings of the international conference on artificial intelligence and soft computing*, pp 379–388
34. Xie Z, Zhang C, Shao X, Lin W, Zhu H (2014) An effective hybrid teaching–learning-based optimization algorithm for permutation flow shop scheduling problem. *Adv Eng Softw* 77:35–47
35. Yenisey MM, Yagmahan B (2014) Multi-objective permutation flow shop scheduling problem: literature review, classification and current trends. *Comput Hum Behav* 45:119–135
36. Yin LJ, Li XY, Gao L, Lu C, Zhang Z (2017) A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem, considering productivity, energy efficiency and noise reduction. *Sustain Comput Inf* 13:15–30
37. Yoshua B, Andrea L, Antoine P (2020) Machine learning for combinatorial optimization: a methodological Tour d’Horizon. *Eur J Oper Res*. <https://doi.org/10.1016/j.ejor.2020.07.063>
38. Yuan H (2019) Spatiotemporal task scheduling for heterogeneous delay-tolerant applications in distributed green data centers. *IEEE/CAA J Autom Sin* 16(4):1686–1697
39. Zheng XL, Wang L (2018) A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem. *IEEE Trans Syst Man Cybern Syst* 48(5):790–800
40. Zhong ZL (2017) Spectral–spatial residual network for hyperspectral image classification: a 3-D deep learning framework. *IEEE Trans Geosci Remote Sens* 56(2):847–858

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations