Scientia Iranica E (2021) 28(2), 950–969

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

A data mining-based solution method for flow shop scheduling problems

B. Özcan[a,*], M. Yavuz[b], and A. Fığlalı[a]

a. *Department of Industrial Engineering, Kocaeli University, Kocaeli, Turkey.*
b. *Department of Information Systems, Statistics and Management Science, University of Alabama, USA.*

Received 19 May 2018; received in revised form 4 March 2019; accepted 3 August 2020

KEYWORDS
Data mining;
Flow shop scheduling;
Heuristic;
Path relinking algorithm;
Optimization.

Abstract. Scheduling is the process of determining where and when to perform manufacturing measures, which is required to conduct activities in a timely, efficient, and cost-effective manner. In this paper, an algorithm is proposed as a solution to the flow shop scheduling problem which holds an important place in the scheduling literature. The path relinking algorithm and data mining are used to solve the flow shop scheduling problem studied here. While DM is used for globally searching the solution space, path relinking is used for local search. Data mining is a method for extracting the embedded information in a cluster that includes implicit information. Path relinking is an algorithm that advances by making binary displacements in order to convert the initial solution to the guiding solution and it is repeated by assigning the best obtained solution within this process to the starting point. The efficiency of the model for Taillard's flow shop scheduling problems was tested. Consequently, it is possible to solve the large-size problem without considerable mathematical background. The obtained results showed that the proposed method comparatively performed as good as other metaheuristic methods.

1. Introduction

Despite the rapid technological development, today, it still takes quite a long time to obtain an optimum solution to large-scale NP-hard problems. To give an example of the optimization field, numerous modeling approaches and solution methods have been developed for NP-hard problems such as the travelling salesman problem, various scheduling problems, and quadratic assignment problem. Optimization algorithms can be basically examined in three groups:

1. Integer programming-based models and methods aimed at presenting exact solutions such as branch-and-bound and dynamic programming;
2. Intuitive models developed specifically for the problems, which do not ensure finding an optimum solution but can give solutions that are close to optimality within a reasonable amount of time;
3. Artificial intelligence-based meta-heuristic models such as simulated annealing, tabu search, artificial neural networks, and genetic algorithms, which can be used for general purposes and give solutions that are either optimum or close to optimality in a reasonable computation time.

In the first group, it can be ensured that the optimal solution is obtained by modeling. However, in general, the solution time exponentially increases

[*] *Corresponding author. Tel.: +90 262 3033313*
Fax: +90 262 3033003;
E-mail address: burcu.ozcan@kocaeli.edu.tr (B. Ozcan)

doi: 10.24200/sci.2020.50995.1957

with problem size. The time required to obtain an optimum solution increases as the problem size grows which exceeds the acceptable limits, meaning that even some moderate-sized practical problems may have impractical solution time durations. The drawback of the heuristic methods is that they are problem specific and ungeneralizable to problem types. Recently, meta-heuristic methods have been widely used in the field of optimization to overcome the above-mentioned challenges. An important advantage of metaheuristics is that they can be implemented without a significant mathematical background [1]. In this study, a new solution method that is free from the above-mentioned issues of metaheuristics and reaches a similar solution quality at competitive times was developed. The proposed method also used a simple model in terms of mathematical infrastructure. Data Mining (DM) analyzes and interprets data to achieve meaningful information stored within large-scale data. In the application of DM in the field of optimization, an attempt is made to reach optimum or near-optimum solutions through hidden and meaningful information involved in the best group of solutions selected from many randomly generated solutions in a short period of time. Increasing the solution performance by applying local search to the best solutions found by the DM approach is also considered as a strategy [2]. One of the issues in the field of production planning, the flowshop scheduling problem, which is known to be NP-hard in terms of difficulty, is to be examined. In the case of small problems, exact solution methods (mathematical programming or enumerative methods such as branch and bound) can find optimal solutions in a short time, but it takes a long time to apply such methods to larger problems that one often confronts in real, practical life. Among the choices suitable for large-sized problems are such approaches as Campbell, Dudek, and Smith (CDS) algorithm, Johnson, Moore, Nawaz, Enscore, Ham (NEH) algorithm as well as Shortest Processing Time (SPT), Earliest Due Date (EDD), First In First Out (FIFO), etc. as priority rules. Another choice of approach that has been widely used for scheduling problems in recent times is meta-heuristic methods such as genetic algorithms, simulated annealing, ant colony optimization, and artificial immune system [3]. In this study, a large number of job sequences will be randomly selected, a certain number of job sequences with the best value in terms of the objective function will be determined, and the DM method will be used to determine the relationships between these solutions. The path relinking algorithm will be applied so that the algorithm developed with DM can become a competitive option. The types of problems applied to the field of optimization of the DM method and the results obtained are summarized. In the literature, it has been observed that the method is applied to small-sized problems and the performance of the method in large-sized problems is investigated. The performance of the proposed model is tested on Taillard's flow shop scheduling problems [4]. It is known that these problems are included in the NP-hard group. The best results obtained so far for these problems are compared with the best results obtained by the model.

2. Flow shop scheduling problems

The flow shop scheduling problem with the total flow time or the completion time objective is flow shop scheduling in the NP-hard class. Therefore, instead of trying to find the optimum result, we focus on high-quality solutions within a reasonable time with heuristic and meta-heuristic methods.

In the n job, m-machine Permutation Flow Shop Scheduling Problem (PFSSP), the processing time of the job i in the j machine $p(i,j)$, work permutation $(\pi_1, \pi_2, \ldots, \pi_n)$, and completion time are defined as follows [5,6]:

$$C\left(\pi_1,1\right) = p\left(\pi_1,1\right),$$

$$C\left(\pi_i,1\right) = C\left(\pi_{i-1},1\right) + p\left(\pi_i,1\right) \quad i=2,\ldots,n,$$

$$C\left(\pi_1,j\right) = C\left(\pi_1,j-1\right) + p\left(\pi_1,j\right) \quad j=2,\ldots,m,$$

$$C\left(\pi_i,j\right) = \max\left\{C\left(\pi_{i-1},j\right), C\left(\pi_i,j-1\right)\right\} + p\left(\pi_i,j\right) \quad i=2,\ldots,n \quad j=2,\ldots,m.$$

The total production time of a product can be defined as follows:

$$C\max\left(\pi\right) = C\left(\pi_n,m\right).$$

Permutation flow-type scheduling problems have the best permutation (π^*) among all permutations.

$$C\max\left(\pi *\right) \leq C\max\left(\pi\right)\pi\varepsilon\Pi.$$

There are many heuristic methods for reducing the completion time, the first of which is Palmer's heuristic [7]. These methods are constructive methods. The best model among these methods is NEH heuristic and the average deviation of this method is over 5%. In recent years, various meta-heuristic methods have been developed for minimizing the completion time in the flow shop scheduling problem. These are simulated annealing, tabu search, genetic algorithms, and hybrid metaheuristics [8]. Meta-heuristic methods generally provide better solution quality than heuristic methods and yield results in a much shorter span of time than exact methods. Heuristic methods are divided into three groups: constructive heuristics [9], improvement heuristics [10], and metaheuristics. Rajendran and Ziegler proposed improving the constructive heuristic algorithm by placing each task in turn [11]. Woo and Yim compared their constructive heuristic algorithm with Rajendran and Ziegler and NEH algorithm and found that their algorithm obtained better results than these two algorithms [12]. Aminnayeri and Naderi proposed a new solution by generalizing the Johnson rule to the flowshop scheduling problem [13]. Liu and Reeves proposed a model that combines multiple constructive heuristics including local search methods. These proposed constructive heuristics indicated the ability to obtain better results [14]. Allahverdi and Aldowaisan showed that the performance could be improved with minor modifications (binary crossover) [15]. Framinan et al. presented an overview of meta-heuristic methods for total flowshop scheduling problems. In addition, they suggested a superior metaheuristic than other heuristics [16]. Jolai et al. showed that metaheuristics employed to deal with the total flow shop scheduling problem was better than constructive and improvement heuristics. In general, metaheuristic showed better results than simpler heuristics, but a longer calculation time was required. They solved the benchmark problems using the flexible flow-type scheduling problem together with the metaheuristic invasive weed optimization method and the response surface method used in parameter optimization [17]. Rajendran and Ziegler proposed two ant colony optimization methods, M-MMAS and Parallel Ant Colony Optimization (PACO), to reduce total flow time and completion time compared with Liu and Reeves algorithm to measure the performance of their methods and they obtained better results in Taillard's benchmark problems [18]. In another study of Rajendran and Ziegler, two ant colony algorithms were improved to obtain better quality solutions [19]. Tasgetiren et al. proposed a particle swarm optimization model and obtained much better results than the algorithms developed by Liu and Reeves and Rajendran and Ziegler with this method. The problem of production and distribution was used together with the mathematical model and improved imperialist competitive algorithm method, and experimental design was used for parameter optimization [20].

3. Data Mining (DM)

DM techniques are grouped into three basic groups in terms of their functions: classification, clustering, and association rules and sequential patterns [21]. Classification ensures that the data are properly separated and partitioned according to the predefined output. The outputs belong to the classification and the dataset is supervised because they are known in advance. Some of these are decision trees, Artificial Neural Networks (ANNs), genetic algorithms, K-nearest neighbor, memory based methods, and regression [22]. Clustering is a technique that groups data according to similarities in the data. The technique is often used as a first step for other applications. Statistical methods such as K-means algorithms or Kohonen network are mostly used. In these methods, the processes are the same. First, each record is compared with the existing clusters, and each record is assigned to the closest cluster and it changes the values that define this cluster. Each time records are reassigned and cluster centers are set [23]. One should consider the significance of association rules and sequential time patterns: Here, the concept that separates the others is the application of time concept. Investigation of the association between objects over a given period is called a sequential pattern [24]. The rules of association are applied in a wide range of areas. The rules of association are the rules that contain objects often appearing together in the same process. The association rules are used in market basket analysis. In this analysis, objects are the products bought by the customer and a transaction (recording) is a single purchase that contains many objects. In the market basket analysis, the objects often taken together are studied. Information on how the rules correspond to each other is collected [25]. The DM technique used in the proposed model can be considered in the classification group. Here, for example, take a problem in which 20 jobs are scheduled; there will be 20 classes representing 20 different positions. While these positions are dependent variables in the target position, the tasks to be assigned to positions are independent variables. The optimization studies in the literature that apply DM to the scheduling domain are summarized in the following.

Koonce et al. studied the workshop scheduling problem. In this study, the rules were determined by considering the process and the load from the optimum job sequences found in the genetic algorithm. In the conclusion and suggestions section, it is mentioned that DM method can be used as an aid to explain complex systems [26]. Koonce and Tsai used a DM algorithm to extract information from a large number of job shop sequencing problems. In the study, job shop scheduling problems proposed by Muth and Thompson (1963) were used [27]. Kumar and Rao solved the block flowshop scheduling problem by using ant colony algorithm and genetic algorithm operators (crossover and mutation) to create new solutions for eight-job four-machine flowshop scheduling. They applied ant colony algorithm and obtained an optimal schedule by calculating the quantity of pheromones for each position [28]. Olafsson and Li proposed a two-step method for the single-machine scheduling problem and argued that new learned rules could go beyond imitation [29]. In another study, Martens et al. underlined that the use of a large number of meta-numerical algorithms derived from nature in conjunction with the popular DM method in recent years would be useful for future research and academical gaps [30]. By defining three objectives for the flexible job shop scheduling problem and using particle swarm optimization and DM techniques, it was argued that this hybrid technique would be a successful approach [31]. Various rule sets were defined using the tabu search method for the workshop type scheduling problem which is determined as the maximum delay target criterion [32]. Petri net was used to propose the job shop scheduling problem model using the following factors: machine time ratio, number of the remaining operations, ratio of processing times of 'operation 1' and 'operation 2', ratio of the remaining processing times of the jobs, and the difference between the waiting times of two operations [33]. In the study of Zahmani et al., simulation and DM were employed together to solve the workshop scheduling problem, and as the objective criterion, the completion time was taken as the basis [34]. Mirshekarian and Šormaz used pearson correlation coefficient of the relationship between job-shop scheduling problem features (Configuration features including overall problem instance, individual operations, jobs, machines, operation slots, machine load, and temporal features which are shortest processing time, longest processing time, much work remaining, least work remaining, first-in-first-out, and optimal makespan [35]. In the study of Makrymanolakis et al., a metaheuristic algorithm was developed to solve the combinatorial optimization problems and follow the DM procedure to select appropriate parameters for different size problems [36]. Senvar et al. investigated large-sized parallel machine scheduling problems and attempted to extract useful knowledge about the domain of these problems. The objective of the study is to provide statistical interpretations using Artificial Neural Networks (ANNs) along with regression analysis and classify the differences between the instances into three clusters in terms of their complexities [37]. Shahzad and Mebarki solved the problem of small-sized workshop scheduling with the Tabu search method and when any two jobs were given by the decision tree method, the tree would predict what job to send first [38]. Wu et al. studied the dynamic customer clustering problem. Qualitative and quantitative rules were employed to respond to instant customer needs. At the same time, they analyzed the distribution and scheduling of third-party logistics with the DM technique [39].

4. Model for flow shop scheduling problems

The first step in the proposed model is design of experiments for determining the best parameter set of the algorithm. Five parameters were determined: initial population (k), random solutions of average table (m), number of locations (l), path relinking algorithm solutions (n), and number of iterations (t). There are three levels for each factor. L27 pattern was used so that the solution could be obtained in a shorter span of time. Table 1 shows the determined factors and levels associated with the design of experiment: initial population (k), number of locations (l), path relinking algorithm solution (n), and number of iterations (t). These levels might be subject to variations in the case of obtaining random solutions of average table (m), because there is no division error in zero as the problem grows.

Table 1. Parameters and levels.

Parameters		Level 1	Level 2	Level 3
	k	500	2500	5000
m	20 jobs	500	2500	5000
	50 jobs	2500	12500	25000
	100 jobs	5000	25000	50000
	l	3	5	8
	n	50	100	150
	t	10	25	50

Twenty-seven different combinations were created for five factors and three levels given in Table 1. Thus, 243 experiments were performed for all combinations (full factorial experiments), while 27 experiments were carried out with a partial experimental design. Thus, a solution was obtained in a shorter span of time. Signal-to-noise (S/N) ratio was calculated based on experimental results. In order to obtain the lowest values, the smallest-is-the-best characteristic was used. The following formula was employed to define this characteristic for analysis:

$$\eta = -10\ Log_{10}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{Y_i^2}\right),$$

where "η" means the (S/N) ratio for "the smallest is the best" characteristic; "y_i" value is obtained in the experiment; "n" denotes the number of repetitions. Given that the objective function for the Tai problems is to minimize the completion time, the experiment is designed with the L27 pattern using "the-smallest-the-best" characteristic. The results are shown in Table 2. The same process is repeated for other problems with different sizes, parameters, levels, and (S/N) ratios (Table 2), while the best levels of the parameters are given in Figure 1.

It was observed that the most important factor was the number of iterations. Figure 1 shows the best levels for five factors.

The parameter level fine tuning values calculated according to signal-noise ratios are shown in Figure 1.

Table 2. Taguchi design of experiment L27.

No.	k	m	l	n	t	S/N	**No.**	k	m	l	n	t	S/N
1	500	500	3	50	10	–62.2588	15	2500	2500	8	50	50	–62.2588
2	500	500	3	50	25	–62.2588	16	2500	5000	3	100	10	–62.2588
3	500	500	3	50	50	–62.2348	17	2500	5000	3	100	25	–62.2575
4	500	2500	5	100	10	–62.2521	18	2500	5000	3	100	50	–62.2267
5	500	2500	5	100	25	–62.2588	19	5000	500	8	100	10	–62.2454
6	500	2500	5	100	50	–62.2335	20	5000	500	8	100	25	–62.2588
7	500	5000	8	150	10	–62.2575	21	5000	500	8	100	50	–62.2160
8	500	5000	8	150	25	–62.2588	22	5000	2500	3	150	10	–62.2588
9	500	5000	8	150	50	–62.1850	23	5000	2500	3	150	25	–62.2588
10	2500	500	5	150	10	–62.2588	24	5000	2500	3	150	50	–62.2093
11	2500	500	5	150	25	–62.2387	25	5000	5000	5	50	10	–62.2588
12	2500	500	5	150	50	–62.2214	26	5000	5000	5	50	25	–62.2548
13	2500	2500	8	50	10	–62.2588	27	5000	5000	5	50	50	–62.1918
14	2500	2500	8	50	25	–62.2588	**min**	**5000**	**5000**	**5**	**150**	**50**	

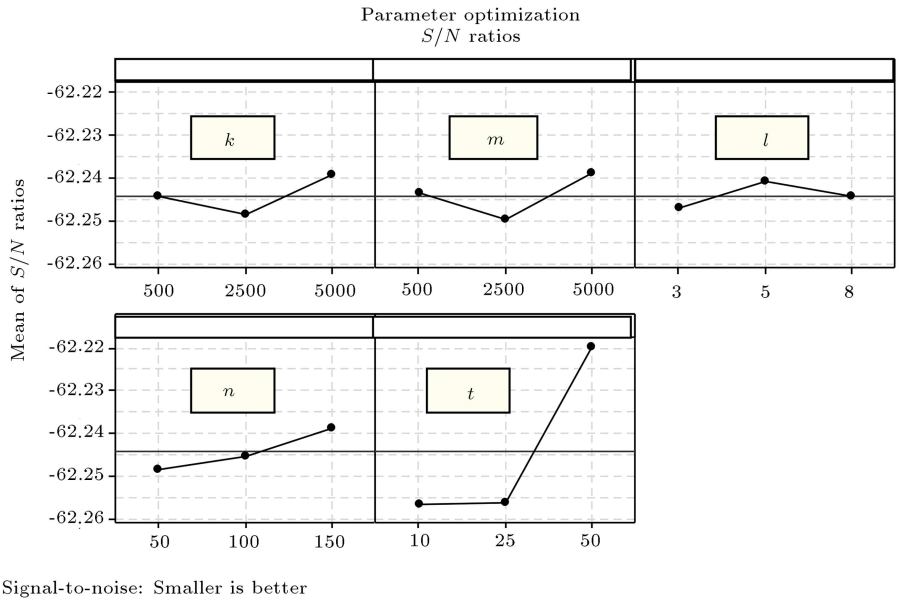


Figure 1. Parameter levels according to S/N ratios.

The minimum values for the 3rd level for initial population, 3rd level for random solutions of average table (m), 2nd level for number of locations (l), 3rd level for the path relinking algorithm solutions (n), and 3rd level for the number of iterations ($A_3B_3C_2D_3E_3$) were obtained. The process followed for Tai01 was applied to other problems, as well. Five trials were conducted for each problem size and the best factor levels were determined, as shown in Table 3.

In this model, the DM method is used to analyze and interpret the data that are stored in order to reach meaningful information stored in large-scale data. In addition, to achieve the best solution, the algorithm

Table 3. The best parameter levels according to the size of the problem.

Problem size	k	m	l	n	t
20 × 5	5000	5000	5	150	50
20 × 10	2500	5000	3	150	50
20 × 20	2500	2500	3	100	50
50 × 5	500	12500	8	150	50
50 × 10	5000	12500	5	150	50
100 × 10	5000	25000	8	150	50

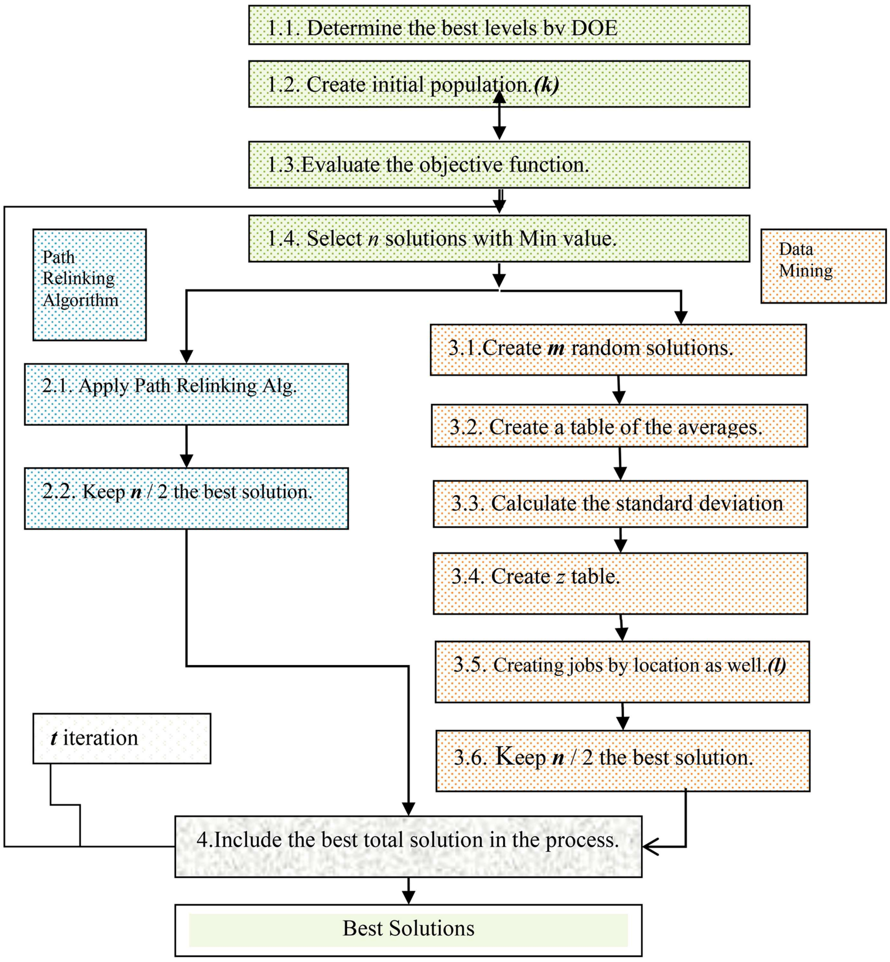

1.1. Determine the best levels bv DOE
1.2. Create initial population.(k)
1.3.Evaluate the objective function.
1.4. Select n solutions with Min value.
Path Relinking Algorithm
Data Mining
2.1. Apply Path Relinking Alg.
2.2. Keep n / 2 the best solution.
3.1.Create m random solutions.
3.2. Create a table of the averages.
3.3. Calculate the standard deviation
3.4. Create z table.
3.5. Creating jobs by location as well.(l)
3.6. Keep n / 2 the best solution.
t iteration
4.Include the best total solution in the process.
Best Solutions

Figure 2. Flowchart of the model proposal.

developed with DM is used to improve the quality of the solution so that it can become a competitive option.

The application of DM in the field of optimization has been researched and it has been found that there are three main works in this area. Although the studies in the literature have outperformed the heuristic ones, their performance remains lower than those of the meta-heuristic ones. It has been stated that the issue remains an open area to develop [40]. For this purpose, a new model is proposed in order to find solutions to large-scale problems and make the obtained results worthy of rivalry with other metaheuristics. In the Matlab 7.0.1 program for the model proposal, the desired number of random solutions is created and the average, standard deviations, and z values of these random rows are calculated according to the positions. The algorithm for the proposed model is shown in Figure 2.

Proposed model algorithm: The proposed model uses DM and path relinking algorithms as a hybrid for optimization of flow-type scheduling problems. When a global search is made in the solution space with DM, an attempt at making a local search is done with path relinking algorithm. The model's effectiveness has been tested on Taillard's flow-type scheduling problems. A large number of randomly generated solutions are used. The average values of $C_{\max}$ obtained when the DM method is used to place each job at a certain position are compared with those of $C_{\max}$ for all the solutions. This is how decisions are made in terms of the suitability of a particular job for placement in a particular position. The path relinking algorithm makes local searches by going through appropriate solutions derived from the DM.

According to flowchart of the model, there are four main steps. These are: 1. Design of experiment, 2. Path relinking algorithm, 3. Data mining, and 4. Hybridize all methods.

1.1 *Create initial population*: The initial population is large, randomized, size of which varies according to the size of the flow shop scheduling problem. K random solutions are created for Taillard problems;

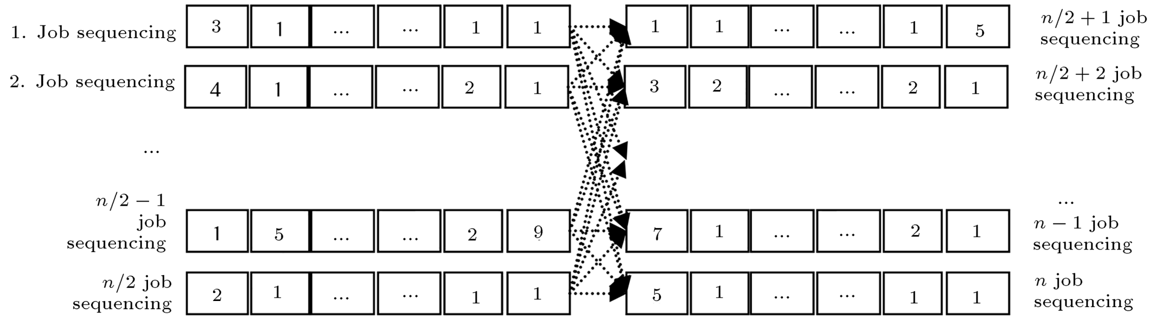
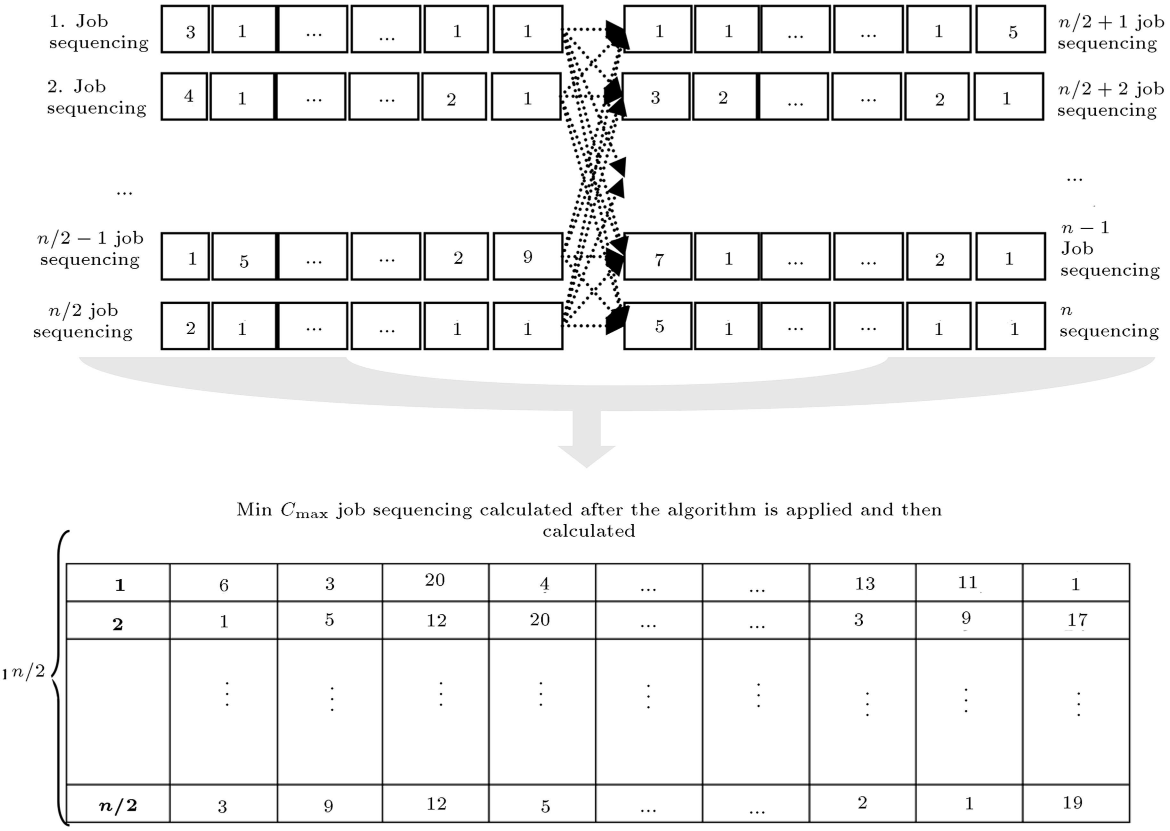


Figure 3. Implementation of the Path relinking Algorithm.



Min $C_{\max}$ job sequencing calculated after the algorithm is applied and then calculated

1	6	3	20	4	...	...	13	11	1
2	1	5	12	20	...	...	3	9	17
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
***n*/2**	3	9	12	5	...	...	2	1	19

Figure 4. Job sequences obtained after the path relinking algorithm.

1.2 *Measure the objective value*: The completion times of the job sequences in the population are calculated;

1.3 *Select* ***n*** *solutions with minimum values*: After generating K random solutions and measuring the objective values, n best solutions with minimum $C_{\max}$ are selected.

2.1 *Apply the path relinking algorithm*: To obtain solutions of higher quality, path relinking is applied to n best solutions. $n/2$ is the source solution and $n/2$ is the target solution. In Figure 3, an example of the implementation of the route combining algorithm is given;

2.2 *Get $n/2$ the best solutions*: After applying path relinking, $n/2$ best solutions are selected. The best $n/2$ solution obtained during the phase is stored to be combined with the DM. Figure 4 shows the job sequence obtained after applying the path relinking algorithm.

3.1 *Create m random solutions*: m random solutions are created. The number of random solutions varies depending on the size of the problem. Considering all combinations for a job ranking problem with a size of 20 × 5, this number is 2.4×10^{18};

3.2 *Create a table of means*: A table of the means according to the positions is created after determining the completion time of each job sequence. Averages of completion times are calculated for each job according to their location in Table 4.

For example, if the 20th job belongs to the first position, the average is calculated given. As shown in Table 4, 2,000,000 random jobs are created and the 20th job arrives at the first position among n20k1 of random 2,000,000 jobs. The completion times of these n20k1 random jobs are calculated and the average of these times is taken, which is shown in Table 5.

These operations were made for 2,000,000 random solutions to the tai01 problem and group averages, standard deviations, and z values were calculated.

Table 6 shows the formulas used for the averages and the results, as given in Table 7;

3.3 *Generate a standard deviation table*: In order to calculate the z table, standard deviations must first be calculated. The group average table is also used to calculate the standard deviation. For example, for the first position (Position 1), the completion times in the case of arriving at Position 1 are subtracted from the average of the completion times for Position 1 of the first task. Then, the squares of the positions are subtracted, and the square root is taken, as shown in Table 8. Table 9 shows the formulas used for standard deviation and the results obtained are given in Table 10;

3.4 *Create z table*: To calculate the z values, the mean and standard deviation must be calculated (z values are calculated to determine the importance of jobs and locations.)

$$z = (\text{group average} - \text{average of all solutions})\,/$$
$$\text{group standard deviation.}$$

Taking + values of z means that when the job is assigned to the specified position, it gets a larger value than the average of all solutions. As this value increases, it gets further away from the average. For example, if the 20th job is in the first position, the z value is 0.622. The average of the group is 1550.63 and the average of the randomly generated job orders is 1515.65. In this case, job 20 should not be assigned to Position 1, as shown in Figure 5. Table 11 shows the formulas used for the z values and the results obtained are given in Table 12;

3.5 *Create a job sequence according to the location*: First, the importance order of the works is put

Table 4. Finding randomized group mean.

	$C_{\max}$	**1**	**2**	**3**	**4**	$\cdots$	$\cdots$	**18**	**19**	**20**
1	**1550**	20	3	5	4	$\cdots$	$\cdots$	13	11	1
2	1611	1	5	12	20	$\cdots$	$\cdots$	3	9	17
3	**1590**	20	3	4	12	$\cdots$	$\cdots$	11	10	5
4	1589	4	9	12	11	$\cdots$	$\cdots$	2	7	20
5	**1600**	20	2	1	14	$\cdots$	$\cdots$	5	10	9
6	**1708**	2	3	15	10	$\cdots$	$\cdots$	12	9	16
	:	:	:	:	:	:	:	:	:	:
1999.998	1594	3	9	12	5	$\cdots$	$\cdots$	2	1	19
1999.999	1603	19	12	20	3	$\cdots$	$\cdots$	7	3	11
2000.000	**1569**	20	19	1	3	$\cdots$	$\cdots$	2	11	17

Table 5. Finding randomized group mean for two million replications.

	$\boldsymbol{k1}$	$\boldsymbol{k2}$	$\cdots$	$\boldsymbol{k20}$
n1	$= (1611 + ...)\,/n1k1$	$= (..........)\,/n1k2$	$\cdots$	$= (1550 + ...)\,/n1k20$
n2	$= (1708 + ...)\,/n2k1$	$= 1600/n2k2$	$\cdots$	$= (..........)\,/n2k20$
n3	$= (1594 + ...)\,/n3k1$	$= (1550 + 1708 + ...)\,/n3k2$	$\cdots$	$= (..........)\,/n3k20$
n4	$= (1589 + ...)\,/n4k1$	$= (..........)\,/n4k2$	$\cdots$	$= (..........)\,/n4k20$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n19	$= (1603 + ...)\,/n19k1$	$= (1569 + ...)\,/n19k2$	$\cdots$	$= (..........)\,/n19k20$
n20	$= (1550 + 1600 + ... + 1569)\,/n20k1$	$= (..........)\,/n20k2$	$\cdots$	$= (..........)\,/n20k20$

Table 6. Formulas of group average

	***k*1**		***k*2**
n1	$\bar{C}_{n1k1=(Cn1k1)1+(Cn1k1)2+\ldots+(Cn1k1)n1k1]/n1k1}$		$\bar{C}_{n1k2=(Cn1k2)1+(Cn1k2)2+\ldots+(Cn1k2)n1k2]/n1k2}$
n2	$\bar{C}_{n2k1=(Cn2k1)1+(Cn2k1)2+\ldots+(Cn2k1)n2k1]/n2k1}$		$\bar{C}_{n2k2=(Cn2k2)1+(Cn2k2)2+\ldots+(Cn2k2)n2k2]/n2k2}$
n3	$\bar{C}_{n3k1=(Cn3k1)1+(Cn3k1)2+\ldots+(Cn3k1)n3k1]/n3k1}$		$\bar{C}_{n3k2=(Cn3k2)1+(Cn3k2)2+\ldots+(Cn3k2)n3k2]/n3k2}$
n4	$\bar{C}_{n4k1=(Cn4k1)1+(Cn4k1)2+\ldots+(Cn4k1)n4k1]/n4k1}$		$\bar{C}_{n4k2=(Cn4k2)1+(Cn4k2)2+\ldots+(Cn4k2)n4k2]/n4k2}$
	...		...
n19	$\bar{C}_{n19k1=(Cn19k1)1+(Cn19k1)2+\ldots+(Cn19k1)n19k1]/n19k1}$		$\bar{C}_{n19k2=(Cn19k2)1+(Cn19k2)2+\ldots+(Cn19k2)n19k2]/n19k2}$
n20	$\bar{C}_{n20k1=(Cn20k1)1+(Cn20k1)2+\ldots+(Cn20k1)n20k1]/n20k1}$		$\bar{C}_{n20k2=(Cn20k2)1+(Cn20k2)2+\ldots+(Cn20k2)n20k2]/n20k2}$
	***k*3**	...	***k*20**
n1	$\bar{C}_{n1k3=(Cn1k3)1+(Cn1k3)2+\cdots+(Cn1k1)n1k1]/n1k1}$	...	$\bar{C}_{n1k20=(Cn1k20)1+(Cn1k20)2+\cdots+(Cn1k20)n1k3]/n1k20}$
n2	$\bar{C}_{n2k3=(Cn2k3)1+(Cn2k3)2+\cdots+(Cn2k3)n2k3]/n2k3}$	...	$\bar{C}_{n2k20=(Cn2k20)1+(Cn2k20)2+\cdots+(Cn2k20)n2k20]/n2k20}$
n3	$\bar{C}_{n3k3=(Cn3k3)1+(Cn3k3)2+\cdots+(Cn3k3)n2k3]/n3k3}$	...	$\bar{C}_{n3k20=(Cn3k20)1+(Cn3k20)2+\cdots+(Cn3k20)n3k20]/n3k20}$
n4	$\bar{C}_{n4k3=(Cn4k3)1+(Cn4k3)2+\cdots+(Cn4k3)n4k3]/n4k3}$	...	$\bar{C}_{n4k20=(Cn4k20)1+(Cn4k20)2+\cdots+(Cn4k20)n4k20]/n4k20}$
	...	...	...
n19	$\bar{C}_{n19k3=(Cn19k3)1+(Cn19k3)2+\cdots+(Cn19k3)n19k3]/n19k3}$	...	$\bar{C}_{n19k20=(Cn19k20)1+(Cn19k20)2+\cdots+(Cn19k20)n19k20]/n19k20}$
n20	$\bar{C}_{n20k3=(Cn20k3)1+(Cn20k3)2+\cdots+(Cn20k3)n20k3]/n20k3}$	...	$\bar{C}_{n20k20=(Cn20k20)1+(Cn20k20)2+\cdots+(Cn20k20)n20k20]/n20k20}$

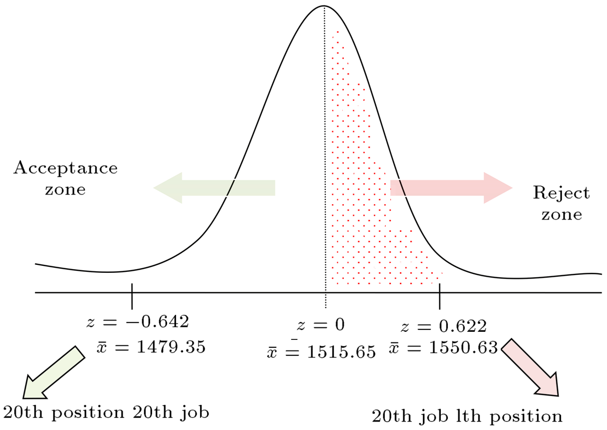


Figure 5. Normal distribution of model.

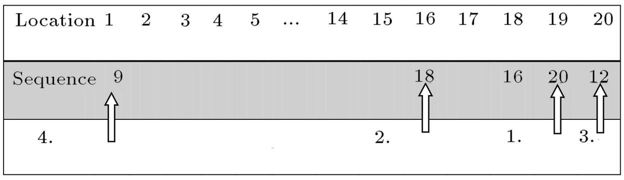


Figure 6. Positioning critical jobs.

forward. Jobs are sorted according to standard deviation values. Thus, the variability of the completion times relative to the positions is measured. For example, as seen in Table 12, job 20 appears to have a larger standard deviation (0.4086 value) than other jobs. This means that job 20 varies greatly from location to location. The twentieth job received a value of 0.622 in the first position, and a value of −0.642 in the twentieth position. In short, the twentieth job moves away from the general average (1515.65) when it comes to its first position and it gets smaller than the average when it reaches the final positions; therefore, the twentieth job should be placed in one of the end positions.

As shown in Table 13, a large number of best solutions have been found with the proposed algorithm for the Tai01 problem, and the twentieth work has been found to be in continuous end positions. In the same way, the ninth job is in the first position among the best job positions achieved and is the fourth in terms of criticality in the z table.

Criteria for all jobs are determined according to the standard deviation and upon completing the ranking task, positioning is done according to the locations. Jobs are assigned to locations starting from the most critical job. For example, five positions are selected, as shown in Table 14. The most critical job will start (20th) and a random assignment will be made to one of the five positions with the lowest z value. For example, the twentieth job is assigned to the nineteenth position, and no further job will be assigned to the nineteenth position. Once the first critical job is completed, the second critical job will be rolled out (job 18). Likewise, one of the five smallest z-values will be randomly selected. For instance, the eighth work is assigned to the sixteenth position. This continues until the standard deviation is the smallest. Figure 6 provides an example of the positioning of critical job orders.

In summary, the job with the largest standard deviation and the ones with the smallest

Table 7. Averages by location for the random solution.

	k1	**k2**	**k3**	**k4**	**k5**	**k6**	**k7**	**k8**	**k9**	**k10**
n1	1511.38	1512.00	1513.33	1513.59	1514.50	1514.93	1515.22	1515.96	1516.02	1516.21
n2	1528.70	1525.37	1523.00	1521.20	1520.01	1518.84	1517.45	1516.13	1515.02	1514.08
n3	1507.67	1508.19	1509.23	1510.54	1511.45	1512.95	1514.08	1515.10	1515.71	1517.38
n4	1514.04	1513.07	1513.63	1514.15	1514.08	1513.95	1514.36	1514.09	1514.08	1513.99
n5	1516.91	1514.67	1514.50	1514.74	1515.40	1516.03	1516.84	1516.78	1517.47	1517.87
n6	1493.99	1498.28	1501.48	1504.73	1507.25	1509.81	1512.53	1515.39	1518.02	1520.32
n7	1514.20	1517.09	1520.27	1521.46	1522.62	1522.69	1522.64	1521.84	1521.22	1519.83
n8	1507.88	1509.82	1511.33	1512.43	1513.11	1514.21	1515.14	1515.58	1516.35	1517.38
n9	1501.68	1501.01	1500.71	1501.42	1502.11	1502.91	1504.67	1506.07	1508.29	1510.35
n10	1532.50	1529.00	1526.65	1525.44	1524.48	1523.58	1522.66	1522.49	1521.46	1520.54
n11	1517.77	1514.40	1511.25	1508.52	1506.14	1504.73	1504.07	1504.05	1504.42	1505.34
n12	1545.06	1540.28	1536.21	1532.32	1528.50	1524.38	1520.13	1516.35	1522.49	1508.61
n13	1493.04	1504.31	1509.83	1513.02	1516.33	1518.88	1521.14	1522.69	1524.04	1524.90
n14	1500.04	1506.15	1509.00	1511.78	1513.14	1514.73	1515.53	1516.93	1517.70	1518.24
n15	1487.77	1493.88	1497.65	1501.00	1503.80	1507.34	1510.18	1512.92	1515.45	1517.53
n16	1529.13	1525.00	1521.42	1518.15	1515.33	1512.87	1509.99	1508.32	1506.62	1505.51
n17	1508.40	1506.57	1506.32	1506.25	1506.54	1506.91	1508.17	1508.75	1509.91	1511.19
n18	1530.17	1527.76	1527.19	1527.00	1526.81	1526.34	1525.56	1525.48	1524.39	1523.16
n19	1522.01	1519.48	1516.97	1515.18	1514.28	1513.19	1512.41	1511.61	1511.35	1511.13
n20	1550.63	1546.50	1542.92	1540.21	1537.11	1533.72	1530.24	1526.40	1523.24	1519.18
	k11	**k12**	**k13**	**k14**	**k15**	**k16**	**k17**	**k18**	**k19**	**k20**
n1	1516.39	1516.47	1516.85	1517.33	1516.61	1516.51	1516.80	1516.08	1516.25	1520.52
n2	1516.39	1516.47	1516.85	1517.33	1516.61	1516.51	1516.80	1516.08	1516.25	1520.52
n3	1513.57	1513.00	1512.00	1511.11	1510.65	1510.31	1509.77	1510.12	1509.85	1512.72
n4	1517.79	1519.00	1519.82	1520.05	1520.47	1520.65	1520.60	1519.58	1518.50	1514.27
n5	1514.08	1514.33	1514.39	1514.56	1514.92	1515.26	1516.41	1517.93	1521.64	1529.95
n6	1517.54	1517.59	1517.73	1517.17	1516.44	1515.34	1514.17	1512.63	1510.91	1512.24
n7	1522.00	1523.62	1525.11	1525.75	1526.35	1526.19	1525.32	1523.86	1519.95	1512.90
n8	1518.27	1517.12	1516.18	1513.70	1512.41	1510.05	1507.89	1505.37	1504.01	1504.16
n9	1518.11	1518.69	1519.00	1518.87	1518.74	1518.39	1518.17	1517.38	1516.24	1515.99
n10	1512.65	1515.26	1517.82	1520.69	1523.80	1527.38	1531.47	1535.25	1540.25	1549.29
n11	1519.30	1517.71	1516.17	1514.21	1511.83	1509.57	1505.79	1501.97	1495.12	1472.29
n12	1506.85	1508.99	1511.75	1514.46	1518.57	1522.72	1527.33	1532.62	1539.73	1549.41
n13	1505.10	1502.25	1499.61	1498.77	1498.59	1512.19	1502.27	1506.85	1513.44	1499.40
n14	1525.75	1525.53	1525.27	1524.16	1522.52	1520.31	1516.88	1512.19	1504.60	1487.18
n15	1518.74	1518.99	1519.27	1519.47	1519.67	1519.22	1518.69	1517.87	1517.44	1520.31
n16	1520.52	1522.60	1524.20	1526.50	1527.31	1528.60	1528.84	1528.66	1528.47	1530.14
n1	1504.82	1504.67	1505.34	1507.15	1509.56	1513.03	1517.88	1524.11	1531.67	1542.16
n17	1513.07	1514.73	1516.35	1518.68	1521.12	1523.37	1525.69	1528.87	1532.31	1539.79
n18	1521.73	1519.80	1517.11	1514.81	1511.42	1507.81	1502.67	1496.88	1488.64	1468.18
n19	1511.59	1511.34	1512.41	1512.83	1513.91	1515.19	1516.94	1519.29	1522.49	1529.35
n20	1515.01	1511.17	1506.59	1502.54	1498.08	1493.66	1489.31	1485.23	1481.61	1479.35
									General mean	1515.65

Table 8. Detailed example of standard deviation table.

	k1	k2	...	k20
n1	$=\sqrt{((1611-1511.38)+(....-1511.38))^2/n1k1}$	$=\sqrt{((....-1512.00))^2/n1k2}$	$\cdots$	$=\sqrt{((1550-1520.52)+(....-1520.52))^2/n1k20}$
n2	$=\sqrt{((1708-1528.70)+(....-1528.70))^2/n2k1}$	$=\sqrt{((1600-1525.37)+(....-1525.37))^2/n2k2}$	$\cdots$	$=\sqrt{(....-1512.72)^2/n1k20}$
n3	$=\sqrt{((1594-1507.67)+(....-1507.67))^2/n3k1}$	$=\sqrt{((1550-1508.19)+(....-1508.19))^2/n2k2}$	$\cdots$	$=\sqrt{(....-1514.27)^2/n3k20}$
n4	$=\sqrt{((1589-1514.04)+(....-1514.04))^2/n4k1}$	$=\sqrt{((....-1513.07))^2/n4k2}$	$\cdots$	$=\sqrt{(....-1529.95)^2/n4k20}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n19	$=\sqrt{((1603-1522.01)+(....-1522.01))^2/n19k1}$	$=\sqrt{((1569-1519.48)+(....-1519.48))^2/n19k2}$	$\cdots$	$=\sqrt{(....-1529.35)^2/n19k20}$
n20	$=\sqrt{((1550-1550.63)+(....-1500.63))^2/n20k1}$	$=\sqrt{((....-1546.50))^2/n20k2}$	$\cdots$	$=\sqrt{(....-1479.35)^2/n20k20}$

z values are placed. Once all jobs are placed according to their criticality, the resulting job order is stored and the fitness value is calculated;

3.6 *Obtain $n/2$ best solutions*: The best $n/2$ solution obtained from work order according to the location is stored for merging with the path relinking algorithm. Path relinking is the generalized version of local search. The solutions are called source solution and initial solution. The starting points of the initial and target solutions represent the desired solutions. Figure 7 shows the path relinking algorithm.

4. *Include the best solution into the process*: $n/2$ solution obtained at 3.5 and $n/2$ solution obtained at 2.2. are combined and sent back to 1.4. The algorithm iterates through the same steps. The stopping criterion is the pre-determined number of iterations. Figure 8 shows n solutions obtained by merging the path relinking and DM algorithms.

Obtained Results: In order to observe the performance of the proposed algorithm, Taillard's flowshop problems were selected and the problems ranging from 20 to 100 jobs were solved, as shown in Table 15.

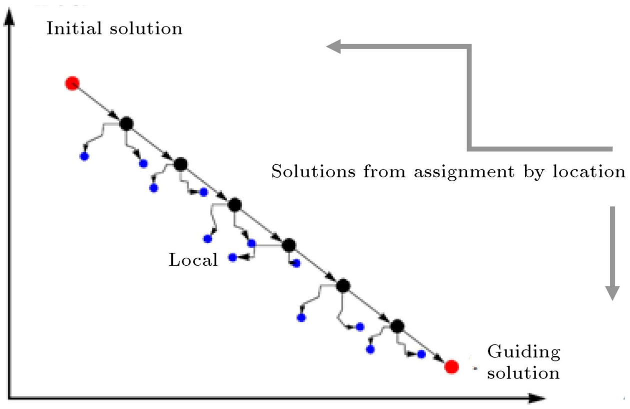


Figure 7. Path relinking algorithm.

5. Conclusion and future studies

Scheduling problems are the optimization problems that belong to the NP-hard class. Numerous different approaches have been developed to achieve the best solutions. These methods include mathematical and non-mathematical methods; the latter methods are divided into two groups of heuristic and meta-heuristic methods. Non-mathematical methods require long solution times and are applicable to finding an optimum solution to small problems. Heuristic methods are

Table 9. Formulas of standard deviation table.

	k1	**k2**
n1	$\sigma_{1k1} = \sqrt{\left(\sum_{l=1}^{n_{1k1}} \left(\left(C_{n_{1k1}}\right)_{j=1}^{n1k1} - \frac{1}{n_{1k1}} \sum_{i=1}^{n_{1k1}} \left(C_{1k1}\right)_i\right)^2 \cdot \frac{1}{n_{1k1}}\right)}$	$\sigma_{1k2}=\sqrt{\left(\sum_{l=1}^{n_{1k2}} \left(\left(C_{n_{1k2}}\right)_{j=1}^{n1k2} - \frac{1}{n_{1k2}} \sum_{i=1}^{n_{1k2}} \left(C_{1k2}\right)_i\right)^2 \cdot \frac{1}{n_{1k2}}\right)}$
n2	$\sigma_{2k1} = \sqrt{\left(\sum_{l=1}^{n_{2k1}} \left(\left(C_{n_{2k1}}\right)_{j=1}^{n2k1} - \frac{1}{n_{2k1}} \sum_{i=1}^{n_{2k1}} \left(C_{2k1}\right)_i\right)^2 \cdot \frac{1}{n_{2k1}}\right)}$	$\sigma_{2k2}=\sqrt{\left(\sum_{l=1}^{n_{2k2}} \left(\left(C_{n_{2k2}}\right)_{j=1}^{n2k2} - \frac{1}{n_{2k2}} \sum_{i=1}^{n_{2k2}} \left(C_{2k2}\right)_i\right)^2 \cdot \frac{1}{n_{2k2}}\right)}$
n3	$\sigma_{3k1} = \sqrt{\left(\sum_{l=1}^{n_{3k1}} \left(\left(C_{n_{3k1}}\right)_{j=1}^{n3k1} - \frac{1}{n_{3k1}} \sum_{i=1}^{n_{3k1}} \left(C_{3k1}\right)_i\right)^2 \cdot \frac{1}{n_{3k1}}\right)}$	$\sigma_{3k2}=\sqrt{\left(\sum_{l=1}^{n_{3k2}} \left(\left(C_{n_{3k3}}\right)_{j=1}^{n3k3} - \frac{1}{n_{3k3}} \sum_{i=1}^{n_{3k3}} \left(C_{3k3}\right)_i\right)^2 \cdot \frac{1}{n_{3k3}}\right)}$
⋮	⋮	⋮
n19	$\sigma_{19K1} = \sqrt{\left(\sum_{l=1}^{n_{19k1}} \left(\left(C_{n_{19k1}}\right)_{j=1}^{n19k1} - \frac{1}{n_{19k1}} \sum_{i=1}^{n_{19k1}} \left(C_{19k1}\right)_i\right)^2 \cdot \frac{1}{n_{19k1}}\right)}$	$\sigma_{19k2}=\sqrt{\left(\sum_{l=1}^{n_{19k2}} \left(\left(C_{n_{19k2}}\right)_{j=1}^{n19k2} - \frac{1}{n_{19k2}} \sum_{i=1}^{n_{19k1}} \left(C_{19k2}\right)_i\right)^2 \cdot \frac{1}{n_{19k2}}\right)}$
n20	$\sigma_{1K20} = \sqrt{\left(\sum_{l=1}^{n_{20k1}} \left(\left(C_{n_{20k1}}\right)_{j=1}^{n20k1} - \frac{1}{n_{20k1}} \sum_{i=1}^{n_{20k1}} \left(C_{20k1}\right)_i\right)^2 \cdot \frac{1}{n_{20k1}}\right)}$	$\sigma_{20k2}=\sqrt{\left(\sum_{l=1}^{n_{20k2}} \left(\left(C_{n_{20k2}}\right)_{j=1}^{n20k2} - \frac{1}{n_{20k2}} \sum_{i=1}^{n_{20k2}} \left(C_{20k2}\right)_i\right)^2 \cdot \frac{1}{n_{20k2}}\right)}$
	...	**k20**
n1	...	$\sigma_{1k20}=\sqrt{\left(\sum_{l=1}^{n_{1k2}^{0}} \left(\left(C_{n_{1k20}}\right)_{j=1}^{n1k20} - \frac{1}{n_{1k20}} \sum_{i=1}^{n_{1k2}} \left(C_{1k20}\right)_i\right)^2 \cdot \frac{1}{n_{1k20}}\right)}$
n2	...	$\sigma_{2k20}=\sqrt{\left(\sum_{l=1}^{n_{2k20}} \left(\left(C_{n_{2k20}}\right)_{j=1}^{n2k20} - \frac{1}{n_{2k20}} \sum_{i=1}^{n_{2k20}} \left(C_{2k20}\right)_i\right)^2 \cdot \frac{1}{n_{2k20}}\right)}$
n3	...	$\sigma_{3k20}=\sqrt{\left(\sum_{l=1}^{n_{3k20}} \left(\left(C_{n_{3k20}}\right)_{j=1}^{n3k20} - \frac{1}{n_{3k3}} \sum_{i=1}^{n_{3k3}} \left(C_{3k20}\right)_i\right)^2 \cdot \frac{1}{n_{3k20}}\right)}$
⋮	⋮	⋮
n19	...	$\sigma_{19k20}=\sqrt{\left(\sum_{l=1}^{n_{19k20}} \left(\left(C_{n_{19k20}}\right)_{j=1}^{n19k20} - \frac{1}{n_{19k20}} \sum_{i=1}^{n_{20k1}} \left(C_{19k20}\right)_i\right)^2 \cdot \frac{1}{n_{19k2}}\right)}$
n20	...	$\sigma_{20k20}=\sqrt{\left(\sum_{l=1}^{n_{20k20}} \left(\left(C_{n_{20k20}}\right)_{j=1}^{n20k1} - \frac{1}{n_{20k20}} \sum_{i=1}^{n_{20k20}} \left(C_{20k20}\right)_i\right)^2 \cdot \frac{1}{n_{20k1}}\right)}$

employed for solving specific problems. Metaheuristic methods are able to approximate local optimum solutions.

There are three basic studies in the literature that have optimized scheduling problems using data mining. Accordingly and in brief, the results of Koonce and Tsai [27] and Koonce et al. [26] emphasized four factors involved in finding the best order in the workshop job

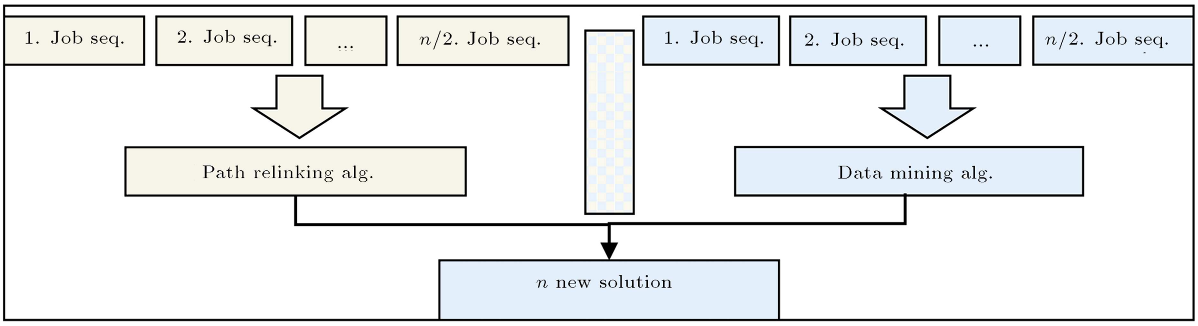


Figure 8. Combining the path relinking and DM algorithms.

Table 10. Standard deviations by position for random solution.

	k1	**k2**	**k3**	**k4**	**k5**	**k6**	**k7**	**k8**	**k9**	**k10**
n1	58.31	59.32	60.00	60.48	61.14	61.43	61.76	61.50	61.33	61.32
n2	60.70	60.50	60.05	59.66	59.52	58.99	58.75	58.93	59.17	59.40
n3	59.19	58.87	58.53	58.58	58.63	58.94	59.31	59.65	60.08	60.74
n4	57.69	59.24	60.47	61.35	61.89	62.11	62.50	62.14	61.81	61.21
n5	58.37	58.62	58.95	58.58	58.67	59.19	59.77	60.43	60.77	61.20
n6	56.57	56.92	57.51	58.17	58.57	58.95	59.68	60.24	60.96	61.30
n7	54.10	57.01	58.71	60.45	62.03	62.97	63.85	64.23	64.30	63.74
n8	58.56	59.20	59.56	60.15	60.02	60.56	60.80	60.89	61.08	61.28
n9	58.14	57.55	57.47	56.83	56.66	56.88	56.78	56.97	57.59	57.94
n10	57.64	58.34	58.11	57.59	57.17	56.95	57.00	57.14	57.51	57.67
n11	58.46	59.36	59.00	58.68	57.39	56.84	55.84	55.21	54.91	55.76
n12	57.71	58.95	59.68	60.00	60.30	59.52	58.99	58.40	59.03	56.73
n13	54.86	57.02	58.44	59.64	60.89	62.03	62.36	62.92	63.19	63.11
n14	56.84	58.41	59.87	60.67	61.56	62.36	62.63	62.83	63.01	62.33
n15	56.17	56.60	56.82	57.32	58.13	58.94	59.62	60.62	60.97	61.20
n16	60.25	60.89	61.11	60.66	60.08	59.14	58.13	57.06	56.50	55.78
n17	59.37	59.47	59.02	58.61	58.29	58.06	58.21	57.98	57.97	58.51
n18	54.14	55.57	56.84	57.76	58.39	58.85	59.34	59.90	59.82	60.21
n19	60.35	60.76	60.31	59.80	59.35	58.89	58.85	58.53	58.60	58.77
n20	56.20	57.64	57.99	58.71	58.38	58.36	57.94	57.71	56.77	56.08

	k11	**k12**	**k13**	**k14**	**k15**	**k16**	**k17**	**k18**	**k19**	**k20**
n1	60.65	60.74	60.56	60.08	59.51	59.36	58.97	58.85	58.17	55.49
n2	59.72	60.04	60.52	60.93	60.99	61.17	60.89	60.35	58.81	55.74
n3	61.11	61.43	61.67	61.49	61.30	60.89	60.16	59.38	58.65	58.14
n4	60.75	60.16	59.53	59.14	58.36	58.48	58.34	58.69	58.35	54.84
n5	61.80	62.20	62.13	62.48	62.26	61.73	60.87	59.94	57.56	53.14
n6	61.46	61.77	61.83	61.49	60.99	60.52	59.58	58.44	56.66	51.00
n7	63.22	62.12	60.83	58.98	57.65	56.20	54.81	54.33	55.26	56.63
n8	60.86	60.81	60.74	60.72	60.23	59.93	59.32	59.00	58.23	56.17
n9	58.46	59.34	59.49	59.90	60.51	60.38	60.55	59.72	58.51	55.41
n10	58.17	58.63	59.13	59.79	60.61	60.89	61.17	61.01	60.12	54.26
n11	56.49	57.67	59.19	60.55	62.16	62.81	63.44	62.45	60.83	55.47
n12	55.53	54.64	54.55	54.89	56.27	57.59	59.82	61.52	61.67	57.98
n13	63.23	62.14	61.08	60.00	58.04	56.51	54.80	52.80	52.06	53.14
n14	61.96	61.34	60.51	59.34	58.42	57.58	56.82	56.34	56.25	55.52
n15	61.44	61.33	61.07	60.41	59.48	58.63	57.63	56.66	55.57	52.41
n16	55.80	55.99	56.85	57.94	59.41	60.93	62.06	62.72	61.91	58.46
n17	58.57	59.10	59.65	60.23	60.97	61.02	61.35	60.82	59.95	56.96
n18	60.24	60.02	59.67	59.32	58.92	58.03	57.31	56.45	55.98	51.54
n19	59.33	59.63	60.11	60.97	61.15	61.36	61.47	61.29	59.81	56.83
n20	55.32	54.45	53.41	52.63	52.24	52.05	52.27	53.58	55.45	56.54

Table 11. z value formulas of model.

	***k*1**	***k*2**
n1	$Z_{1k1}=\left(\bar{C}_{n_{1k1}}-(\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n)\right)/\sigma_{1k1}$	$Z_{1k2}=\left(\bar{C}_{n_{1k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{1k2}$
n2	$Z_{2k1}=\left(\bar{C}_{n_{2k1}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{2k1}$	$Z_{2k2}=\left(\bar{C}_{n_{2k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{2k2}$
n3	$Z_{3k1}=\left(\bar{C}_{n_{3k1}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{3k1}$	$Z_{3k2}=\left(\bar{C}_{n_{3k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{3k2}$
n4	$Z_{4k1}=\left(\overline{C}_{n_{4k1}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{4k1}$	$Z_{4k2}=\left(\bar{C}_{n_{4k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{4k2}$
⋮	⋮	⋮
n19	$Z_{19k1}=\left(\overline{C}_{n_{19k1}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{19k1}$	$Z_{19k2}=\left(\overline{C}_{n_{19k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{19k2}$
n20	$Z_{20k1}=\left(\overline{C}_{n_{20k1}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{20k1}$	$Z_{20k2}=\left(\bar{C}_{n_{20k2}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{20k2}$

	···	***k*2**
n1	⋮	$Z_{1k20}=\left(\overline{C}_{n_{1k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{1k20}$
n2	⋮	$Z_{2k20}=\left(\overline{C}_{n_{2k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{2k20}$
n3	⋮	$Z_{3k20}=\left(\overline{C}_{n_{3k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{3k20}$
n4	⋮	$Z_{4k20}=\left(\bar{C}_{n_{4k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{4k20}$
:	:	:
n19	⋮	$Z_{19k20}=\left(\bar{C}_{n_{19k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{19k20}$
n20	⋮	$Z_{20k20}=\left(\bar{C}_{n_{20k20}}-\sum_{j=1}^{20}\sum_{i=1}^{20}C_{nikj}/n\right)/\sigma_{20k20}$

scheduling problem and formed a number of rules by classifying these factors. Santos (2006) presented some alternatives to improve the performance of development algorithms on a known problem such as vehicle problem. Although the studies in the literature perform better than the heuristic ones, they, reportedly, have lower performance than the meta-heuristic ones and the above-mentioned authors were of the opinion that the subject was an open field for development.

For this purpose, an algorithm including a DM

Table 12. Finding z value for statistical solutions.

	k1	**k2**	**k3**	**k4**	**k5**	**k6**	**k7**	**k8**	**k9**	**k10**
n1	**–0.073**	**–0.062**	**–0.039**	**–0.034**	**–0.019**	**–0.012**	**–0.007**	0.005	0.006	0.009
n2	**–0.073**	**–0.062**	**–0.039**	**–0.034**	**–0.019**	**–0.012**	**–0.007**	0.005	0.006	0.009
n3	0.215	0.161	0.122	0.093	0.073	0.054	0.031	0.008	**–0.011**	**–0.026**
n3	**–0.135**	**–0.127**	**–0.110**	**–0.087**	**–0.072**	**–0.046**	**–0.026**	**–0.009**	0.001	0.028
n4	**–0.028**	**–0.043**	**–0.033**	**–0.024**	**–0.025**	**–0.027**	**–0.021**	**–0.025**	**–0.025**	**–0.027**
n5	0.022	**–0.017**	**–0.020**	**–0.016**	**–0.004**	0.007	0.020	0.019	0.030	0.036
n6	**–0.383**	**–0.305**	**–0.246**	**–0.188**	**–0.143**	**–0.099**	**–0.052**	**–0.004**	0.039	0.076
n7	**–0.027**	0.025	0.079	0.096	0.112	0.112	0.110	0.096	0.087	0.066
n8	**–0.133**	**–0.099**	**–0.072**	**–0.053**	**–0.042**	**–0.024**	**–0.008**	**–0.001**	0.011	0.028
n9	**–0.240**	**–0.254**	**–0.260**	**–0.250**	**–0.239**	**–0.224**	**–0.193**	**–0.168**	**–0.128**	**–0.091**
n10	0.292	0.229	0.189	0.170	0.155	0.139	0.123	0.120	0.101	0.085
n11	0.036	**–0.021**	**–0.074**	**–0.121**	**–0.166**	**–0.192**	**–0.207**	**–0.210**	**–0.204**	**–0.185**
n12	0.510	0.418	0.344	0.278	0.213	0.147	0.076	0.012	0.116	**–0.124**
n13	**–0.412**	**–0.199**	**–0.100**	**–0.044**	0.011	0.052	0.088	0.112	0.133	0.147
n14	**–0.275**	**–0.163**	**–0.111**	**–0.064**	**–0.041**	**–0.015**	**–0.002**	0.020	0.033	0.042
n15	**–0.496**	**–0.385**	**–0.317**	**–0.255**	**–0.204**	**–0.141**	**–0.092**	**–0.045**	**–0.003**	0.031
n16	0.224	0.154	0.095	0.041	**–0.005**	**–0.047**	**–0.097**	**–0.128**	**–0.160**	**–0.182**
n17	**–0.122**	**–0.153**	**–0.158**	**–0.160**	**–0.156**	**–0.151**	**–0.129**	**–0.119**	**–0.099**	**–0.076**
n18	0.268	0.218	0.203	0.197	0.191	0.182	0.167	0.164	0.146	0.125
n19	0.105	0.063	0.022	**–0.008**	**–0.023**	**–0.042**	**–0.055**	**–0.069**	**–0.073**	**–0.077**
n20	0.622	0.535	0.470	0.418	0.368	0.310	0.252	0.186	0.134	0.063

	k11	**k12**	**k13**	**k14**	**k15**	**k16**	**k17**	**k18**	**k19**	**k20**
n1	0.012	0.013	0.020	0.028	0.016	0.015	0.020	0.007	0.010	0.088
n2	**–0.035**	**–0.044**	**–0.060**	**–0.074**	**–0.082**	**–0.087**	**–0.097**	**–0.092**	**–0.099**	**–0.052**
n3	0.035	0.055	0.068	0.072	0.079	0.082	0.082	0.066	0.049	**–0.024**
n4	**–0.026**	**–0.022**	**–0.021**	**–0.018**	**–0.013**	**–0.007**	0.013	0.039	0.103	0.261
n5	0.031	0.031	0.033	0.024	0.013	**–0.005**	**–0.024**	**–0.050**	**–0.082**	**–0.064**
n6	0.103	0.129	0.153	0.164	0.175	0.174	0.162	0.140	0.076	**–0.054**
n7	0.042	0.024	0.009	**–0.033**	**–0.056**	**–0.100**	**–0.141**	**–0.189**	**–0.211**	**–0.203**
n8	0.040	0.050	0.055	0.053	0.051	0.046	0.043	0.029	0.010	0.006
n9	**–0.051**	**–0.007**	0.037	0.084	0.135	0.194	0.261	0.328	0.421	0.607
n10	0.063	0.035	0.009	**–0.024**	**–0.063**	**–0.100**	**–0.161**	**–0.224**	**–0.342**	**–0.799**
n11	**–0.156**	**–0.115**	**–0.066**	**–0.020**	0.047	0.113	0.184	0.272	0.396	0.609
n12	**–0.190**	**–0.245**	**–0.294**	**–0.308**	**–0.303**	**–0.060**	**–0.224**	**–0.143**	**–0.036**	**–0.280**
n13	0.160	0.159	0.157	0.142	0.118	0.083	0.022	**–0.065**	**–0.212**	**–0.536**
n14	0.050	0.054	0.060	0.064	0.069	0.062	0.054	0.039	0.032	0.084
n15	0.079	0.113	0.140	0.180	0.196	0.221	0.229	0.230	0.231	0.276
n16	**–0.194**	**–0.196**	**–0.181**	**–0.147**	**–0.103**	**–0.043**	0.036	0.135	0.259	0.454
n17	**–0.044**	**–0.016**	0.012	0.050	0.090	0.126	0.164	0.217	0.278	0.424
n18	0.101	0.069	0.024	**–0.014**	**–0.072**	**–0.135**	**–0.226**	**–0.332**	**–0.482**	**–0.921**
n19	**–0.068**	**–0.072**	**–0.054**	**–0.046**	**–0.028**	**–0.007**	0.021	0.059	0.114	0.241
n20	**–0.012**	**–0.082**	**–0.170**	**–0.249**	**–0.336**	**–0.422**	**–0.504**	**–0.568**	**–0.614**	**–0.642**

Table 13. Best solution alternatives.

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	$C_{\max}$
Best solutions	**1**	3	17	15	8	9	6	5	14	16	7	11	13	18	19	1	4	2	10	20	12	1278
	2	9	17	15	8	3	6	14	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	3	9	17	15	8	3	6	16	14	5	7	11	13	19	1	18	4	2	10	20	12	1278
	4	9	17	15	8	3	14	6	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	5	9	17	15	8	3	14	16	6	5	7	11	13	19	1	18	4	2	10	20	12	1278
	6	9	17	15	8	3	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	7	9	17	15	8	3	6	16	14	5	7	11	13	1	19	18	4	2	10	20	12	1278
	8	9	17	15	8	3	14	6	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	9	9	17	15	8	3	14	16	6	5	7	11	13	1	19	18	4	2	10	20	12	1278
	10	3	17	15	8	9	6	14	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	11	3	17	15	8	9	6	16	14	5	7	11	13	19	1	18	4	2	10	20	12	1278
	12	3	17	15	8	9	14	6	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	13	3	17	15	8	9	14	16	6	5	7	11	13	19	1	18	4	2	10	20	12	1278
	14	3	17	15	8	9	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	15	3	17	15	8	9	6	16	14	5	7	11	13	1	19	18	4	2	10	20	12	1278
	16	3	17	15	8	9	14	6	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	17	3	17	15	8	9	6	5	14	16	7	11	13	18	19	1	4	2	10	20	12	1278
	18	3	17	15	8	9	14	16	6	5	7	11	13	1	19	18	4	2	10	20	12	1278
	19	3	8	17	15	9	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	20	9	15	17	8	3	14	6	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	21	9	15	8	17	3	14	16	6	5	7	11	13	19	1	18	4	2	10	20	12	1278
	22	9	8	17	15	3	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	23	9	17	15	8	3	6	14	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	24	9	15	17	8	3	14	6	16	5	7	11	13	19	1	18	4	2	10	20	12	1278
	25	9	15	8	17	3	14	16	6	5	7	11	13	19	1	18	4	2	10	20	12	1278
	26	9	8	17	15	3	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	27	9	8	17	15	3	6	14	16	5	7	11	13	1	19	18	4	2	10	20	12	1278
	28	3	17	15	8	9	6	16	14	5	11	13	7	19	1	18	4	2	10	20	12	1278
	29	3	17	15	8	9	14	6	16	5	13	11	7	19	1	18	4	2	10	20	12	1278
	30	3	17	15	8	9	14	6	16	11	13	5	7	1	19	18	4	2	10	20	12	1278
	31	3	17	15	8	9	14	6	16	7	11	5	13	1	19	18	4	2	10	20	12	1278

method for solving the flow shop scheduling problems, which plays an important role in scheduling problems in the study, was proposed. To improve the quality of the obtained solutions, local search was performed using the path relinking algorithm. The effectiveness of the model was tested on Taillard's flowshop scheduling problems.

The number of jobs and machines in working problems is 20 and 5, respectively. In order to observe the performance of the proposed algorithm, Taillard flow shop scheduling problems were selected, the number of jobs varying between 20 and 100 problems were solved. Optimum solutions have been found in a large number of different job orders. In this study, the DM technique was used outside the usual format in the literature and it was possible to solve large-scale problems without important mathematical infrastructure.

With the proposed method, 30 problems within 3 different sizes (20×5, 50×10, and 100×10) are solved. The optimum solution was found in 10 out of 30 problems, and it was performed with less than 1% deviation

Table 14. Assignment by location.

Sequence	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
5	0.022	-0.017	-0.020	-0.016	-0.004	0.007	0.020	0.019	0.030	0.036	
1	-0.073	-0.062	-0.039	-0.034	-0.019	-0.012	-0.007	0.005	0.006	0.009	
8	-0.133	-0.099	-0.072	-0.053	-0.042	-0.024	-0.008	-0.001	0.011	0.028	
4	-0.028	-0.043	-0.033	-0.024	-0.025	-0.027	-0.021	-0.025	-0.025	-0.027	
3	-0.135	-0.127	-0.110	-0.087	-0.072	-0.046	-0.026	-0.009	0.001	0.028	
19	0.105	0.063	0.022	-0.008	-0.023	-0.042	-0.055	-0.069	-0.073	-0.077	
14	-0.275	-0.163	-0.111	-0.064	-0.041	-0.015	-0.002	0.020	0.033	0.042	
2	0.215	0.161	0.122	0.093	0.073	0.054	0.031	0.008	-0.011	-0.026	
7	-0.027	0.025	0.079	0.096	0.112	0.112	0.110	0.096	0.087	0.066	
17	-0.122	-0.153	-0.158	-0.160	-0.156	-0.151	-0.129	-0.119	-0.099	-0.076	
6	-0.383	-0.305	-0.246	-0.188	-0.143	-0.099	-0.052	-0.004	0.039	0.076	
16	0.224	0.154	0.095	0.041	-0.005	-0.047	-0.097	-0.128	-0.160	-0.182	
13	-0.412	-0.199	-0.100	-0.044	0.011	0.052	0.088	0.112	0.133	0.147	
11	0.036	-0.021	-0.074	-0.121	-0.166	**-0.192**	**-0.207**	**-0.210**	**-0.204**	**-0.185**	
15	-0.496	**-0.385**	**-0.317**	**-0.255**	**-0.204**	**-0.141**	-0.092	-0.045	-0.003	0.031	
10	0.292	0.229	0.189	0.170	0.155	0.139	0.123	0.120	0.101	0.085	
9	**-0.240**	**-0.254**	**-0.260**	**-0.250**	**-0.239**	-0.224	-0.193	-0.168	-0.128	-0.091	
12	0.510	0.418	0.344	0.278	0.213	0.147	0.076	0.012	0.116	-0.124	
18	0.268	0.218	0.203	0.197	0.191	0.182	0.167	0.164	0.146	0.125	
20	0.622	0.535	0.470	0.418	0.368	0.310	0.252	0.186	0.134	0.063	
Sequence	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	*SS*
5	0.031	0.031	0.033	0.024	0.013	–0.005	–0.024	–0.050	–0.082	–0.064	0.0342
1	0.012	0.013	0.020	0.028	0.016	0.015	0.020	0.007	0.010	0.088	0.0346
8	0.040	0.050	0.055	0.053	0.051	0.046	0.043	0.029	0.010	0.006	0.0544
4	–0.026	–0.022	–0.021	–0.018	–0.013	–0.007	0.013	0.039	0.103	0.261	0.0690
3	0.035	0.055	0.068	0.072	0.079	0.082	0.082	0.066	0.049	–0.024	0.0738
19	–0.068	–0.072	–0.054	–0.046	–0.028	–0.007	0.021	0.059	0.114	0.241	0.0821
14	0.050	0.054	0.060	0.064	0.069	0.062	0.054	0.039	0.032	0.084	0.0911
2	–0.035	–0.044	–0.060	–0.074	–0.082	–0.087	–0.097	–0.092	–0.099	–0.052	0.0925
7	0.042	0.024	0.009	–0.033	–0.056	–0.100	–0.141	–0.189	–0.211	–0.203	0.1106
17	–0.044	–0.016	0.012	0.050	0.090	0.126	0.164	0.217	0.278	0.424	0.1689
6	0.103	0.129	0.153	0.164	0.175	0.174	0.162	0.140	0.076	–0.054	0.1731
16	–0.194	–0.196	–0.181	–0.147	–0.103	–0.043	0.036	0.135	0.259	0.454	0.1785
13	0.160	0.159	0.157	0.142	0.118	0.083	0.022	–0.065	–0.212	–0.536	0.1967
11	–0.156	–0.115	–0.066	–0.020	0.047	0.113	0.184	0.272	0.396	0.609	0.2226
15	0.079	0.113	0.140	0.180	0.196	0.221	0.229	0.230	0.231	0.276	0.2324
10	0.063	0.035	**0.009**	–**0.024**	–**0.063**	–0.100	–**0.161**	–**0.224**	–0.342	–0.799	0.2454
9	–0.051	–0.007	0.037	0.084	0.135	0.194	0.261	0.328	0.421	0.607	0.2553
12	–0.190	–**0.245**	–**0.294**	–**0.308**	–**0.303**	–0.060	–0.224	–0.143	–0.036	–**0.280**	0.2570
18	0.101	0.069	0.024	–**0.014**	–0.072	–0.135	–**0.226**	–**0.332**	–0.482	–**0.921**	0.2940
20	–0.012	–0.082	–0.170	–0.249	–0.336	–**0.422**	–**0.504**	–**0.568**	–0.614	–**0.642**	0.4086

Table 15. Obtained results.

	PS	**BKS**	**LB**	**NEH**	**DM+PR**	**Deviation**
1	20X5	1278	1278	1286	**1278**	**0.00%**
2	20X5	1359	1359	1365	**1359**	**0.00%**
3	20X5	1081	1081	1159	1088	0.65%
4	20X5	1293	1293	1325	1301	0.62%
5	20X5	1235	1235	1305	**1235**	**0.00%**
6	20X5	1195	1195	1228	**1195**	**0.00%**
7	20X5	1239	1239	1278	**1239**	**0.00%**
8	20X5	1206	1206	1223	**1206**	**0.00%**
9	20X5	1230	1230	1291	1233	0.24%
10	20X5	1108	1108	1151	1108	**0.00%**
	PS	**BKS**	**LB**	**NEH**	**DM+PR**	**Deviation**
1	50X10	2991	2991	3135	3000	0.30%
2	50X10	2867	2867	3032	2883	0.56%
3	50X10	2839	2839	2986	2857	0.63%
4	50X10	3063	3063	3198	3083	0.65%
5	50X10	2976	2976	3160	2998	0.74%
6	50X10	3006	3006	3178	3027	0.69%
7	50X10	3093	3093	3277	3107	0.45%
8	50X10	3037	3037	3123	3043	0.20%
9	50X10	2897	2897	3002	2903	0.21%
10	50X10	3065	3065	3257	3084	0.62%
	PS	**BKS**	**LB**	**NEH**	**DM+PR**	**Deviation**
1	100X10	5770	5770	5846	5787	0.29%
2	100X10	5349	5349	5453	5349	**0.00%**
3	100X10	5676	5676	5824	5676	**0.00%**
4	100X10	5781	5781	5929	5791	0.17%
5	100X10	5467	5467	5679	5468	0.02%
6	100X10	5303	5303	5375	5306	0.06%
7	100X10	5595	5595	5704	5598	0.05%
8	100X10	5617	5617	5760	5626	0.16%
9	100X10	5871	5871	6032	5878	0.12%
10	100X10	5845	5845	5918	5845	**0.00%**

from optimum value for all cases. In summary, the results show that the proposed method can compete with other metaheuristic methods.

References

1. Information Resources Management Association, *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, IGI Global Publishing, USA (2016).
2. Thuraisingham, B., *Data Mining: Technologies, Techniques, Tools, and Trends*, CRC Press, Printed in United States of America (2014).
3. Framinan, J.M., Leisten, R., and García, R.R., *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*, Springer Science & Business Media (2014).
4. URL-1: http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ (Visit date: 20.04.2018).
5. Wu, M., Liu, K., and Yang, H. "Supply chain production and delivery scheduling based on data mining", *Cluster Computing*, **22**, pp. 8541–8552 (2019). https://doi.org/10.1007/s10586-018-1894-8
6. Santos, H.G., Ochi Marinho, L.S., and Drummond, E.H. "Combining an evolutionary algorithm with data mining to solve a single-vehicle routing problem", *Neurocomputing*, **70**, pp. 70–77 (2006).
7. Gupta, A. and Kumar, S. "Flow shop scheduling decisions through Techniques for Order Preference by Similarity to an Ideal Solution (TOPSIS)", *International Journal of Production Management and Engineering*, **4**(2), pp. 43–52 (2016). DOI: 10.4995/ijpme.2016.4102
8. Reeves, C.R. and Yamada, T. "Genetic algorithm, path relinking and the flowshop sequencing problem", *Evolutionary Computation*, **6**, pp. 45–60 (1998).
9. Wang, C., Chu, C., and Proth, J.M. "Heuristic approaches for n=m=F==PCi scheduling problems", *European Journal of Operational Research*, **96**, pp. 636–644 (1997).
10. Ho, J.C. "Flowshop sequencing with mean flowtime objective", *European Journal of Operational Research*, **81**(3), pp. 571–578 (1995).
11. Rajendran, C. and Ziegler, H. "An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs", *European Journal of Operational Research*, **103**, pp. 129–138 (1997).
12. Woo, D.S. and Yim, H.S. "A heuristic algorithm for mean flowtime objective in flowshop scheduling", *Computers and Operational Research*, **25**, pp. 175–182 (1998).
13. Aminnayeri, M. and Naderi, B. "Novel properties along with solution methods for permutation flowshop scheduling", *Scientia Iranica, Transaction E, Industrial Engineering*, **23**(5), pp. 2261–2276 (2016).
14. Liu, J. and Reeves, C.R. "Constructive and composite heuristic solutions to the P==PCi scheduling problem", *European Journal of Operational Research*, **132**, pp. 439–452 (2001).
15. Allahverdi, A. and Aldowaisan, T. "New heuristics to minimize total completion time in m-machine flowshops", *International Journal of Production Economics*, **77**, pp. 71–83 (2002).
16. Framinan, J.M., Leisten, R., and Ruiz-Usano, R., "Comparison of heuristics for minimisation in permutation flowshops", *Computers and Operations Research*, **32**, pp. 1237–1254 (2005).
17. Jolai, F., Tavakkoli-Moghaddam, R., Rabiee, M., and Gheisariha, E. "An enhanced invasive weed ptimization for makespan minimization in a flexible flowshop scheduling problem", *Scientia Iranica, Transaction E, Industrial Engineering*, **21**(3), pp. 1007–1020 (2014).

18. Rajendran, C. and Ziegler, H. "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs", *European Journal of Operational Research*, **155**, pp. 426–438 (2004).

19. Rajendran, C. and Ziegler, H. "Two ant-colony algorithms for minimizing total flowtime in permutation flowshops", *Computer and Industrial Engineering*, **48**, pp. 789–797 (2005).

20. Tasgetiren, F., Liang, Y.C., Sevkli, M., and Gencyilmaz, G. "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem", *European Journal of Operational Research*, **177**, pp. 1930–1947 (2007).

21. Mohammadi, S., Cheraghalikhani, A., and Ramezanian, R. "A joint scheduling of production and distribution operations in a flow shop manufacturing system", *Scientia Iranica*, **25**(2), pp. 911–930 (2018).

22. Akpınar, H. "Information discovery and data mining in databases", *Istanbul University Business School Journal*, **29**, pp. 1–22 (2000).

23. Hui, S. and Jha, G. "Application data mining for customer service support", *Information and Management*, **38**, pp. 1–13 (2000).

24. Goebel, M. and Gruenwald, L. "A survey of data mining and knowledge discovery software tools", *ACM SIGKDD Explorations Newsletter*, **1**, pp. 20–33 (1999).

25. Gönülol, S. "A data mining based intuitive approach for the travel salesman problem", M.Sc. Thesis, Kocaeli University, Institute of Science, Kocaeli (2009).

26. Koonce, D., Fang, C.H., and Tsai, S.C. "A data mining tool for learning from manufacturing systems", *Computers Industrial Engineering*, **33**, pp. 27–30 (1997).

27. Koonce, D.A. and Tsai, S.C. "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule", *Computers & Industrial Engineering*, **38**, pp. 361–374 (2000).

28. Kumar, S. and Rao, C.S.P. "Application of antcolony, genetic algorithm and data mining-based techniques for scheduling", *Robotics and Computer-Integrated Manufacturing*, **25**, pp. 901–908 (2009).

29. Olafsson, S. and Li, X. "Learning effective new single machine dispatching rules from optimal scheduling data", *International Journal of Production Economics*, **128**(1), pp. 118–126 (2010).

30. Martens, D., Baesens, B., and Fawcett, T. "Editorial survey: Swarm intelligence for data mining", *Machine Learning*, **82**(1), pp. 1–42 (2011).

31. Karthikeyan, S., Asokan, P., Nickolas, S., and Page, T. "Solving flexible job-shop scheduling problem using hybrid particle swarm optimisation algorithm and data mining", *International Journal of Manufacturing Technology and Management*, **26**(1), pp. 81–103 (2012).

32. Shahzad, A. and Mebarki, N. "Data mining based job dispatching using hybrid simulation-optimization approach for shop scheduling problem", *Engineering Applications of Artificial Intelligence*, **25**(6), pp. 1173–1181 (2012).

33. Wang, C.L., Rong, G., Weng, W., and Feng, Y.P. "Mining scheduling knowledge for job shop scheduling problem", *FAC-PapersOnLine*, **48**(3), pp. 800–805 (2015).

34. Zahmani, M.H., Atmani, B., Bekrar, A., and Aissani, N. "A real time data mining rules selection model for the job shop scheduling problem", *CIE45 Proceedings*, Metz, France (2015).

35. Mirshekarian, S. and Šormaz D.N. "Correlation of job-shop scheduling problem features with scheduling efficiency", *Expert Systems With Applications*, **62**, pp. 131–147 (2016).

36. Makrymanolakis, N., Marinaki, M., and Marinakis, Y. "Data mining parameters' selection procedure applied to a multi-start local search algorithm for the permutation flow shop scheduling problem", *Computational Intelligence (SSCI), IEEE Symposium Series* on (2016).

37. Senvar, O., Yalaoui, F., Dugardin, F., Lara, A.F.B. "Data mining approaches for the methods to minimize total tardiness in parallel machine scheduling problem", *IFAC Conference on Manufacturing Modelling, Management and Control*, Troyes, France (2016).

38. Shahzad, A. and Mebarki, N. "Learning dispatching rules for scheduling: A synergistic view comprising decision trees, tabu search and simulation", *Computers*, **5**(1), pp. 1–16 (2016).

39. Wu, M., Liu, K., and Yang, H. "Supply chain production and delivery scheduling based on data mining", *Cluster Computing*, **22**, pp. 8541–8552 (2019).

40. Poursabzi, O., Mohammadi, M., and Naderi, B. "An improved model and heuristic for capacitated lot-sizing and scheduling in job shop problems", *Scientia Iranica*, **25**(6), pp. 3667–3684 (2018).

Biographies

Burcu Özcan is currently an Associate Professor at the Department of Industrial Engineering in the Faculty of Engineering at Kocaeli University in Kocaeli, Turkey. The author received her BSc and MSc from Kocaeli University. She also conducted her PhD studies at the same university. The main research areas of interest include data mining, scheduling, production planning and control, multi-criteria decision-making, and statistical analysis.

Mesut Yavuz is an Associate Professor of Operations Management in Culverhouse College of Commerce and Business Administration. As native of Turkey, he received his BS and MS degrees from Istanbul Technical University, and later his PhD from the University of

Florida. Upon teaching at the University of Florida and Shenandoah University for 9 years, Dr. Yavuz joined Culverhouse in 2014. His research lies in the broad areas of manufacturing scheduling, sustainable transportation, and energy. His research has been supported by public and private agencies and published in journals including Manufacturing and Service Operations Management and Transportation Science.

Alpaslan Fığlalı is currently a Professor of Industrial Engineering at Kocaeli University in Turkey. He received his BS and MS degrees from Istanbul Technical University, Mechanical Engineering Department and later his PhD from the Industrial Engineering Department of the same University. His research topics are decision theory, optimization, and supply chain management.