



Minimizing the resource consumption of heterogeneous batch-processing machines using a copula-based estimation of distribution algorithm

Chuang Liu^a, Huaping Chen^a, Rui Xu^{b,*}, Yu Wang^a

^a School of Management, University of Science and Technology of China, Hefei 230026, China

^b School of Business, Hohai University, Nanjing 210098, China

HIGHLIGHTS

- A flow shop scheduling problem on batch-processing machines is considered.
- The objective is to minimize resource consumption under the deadline constraint.
- An attempt is made to solve the scheduling problem by using the copula-based EDA.
- An advanced dynamical adjustment to correction algorithm is developed.
- Computational results show that the copula-based EDA outperforms GA, PSO and MIMIC_c.

ARTICLE INFO

Article history:

Received 9 May 2017

Received in revised form 2 August 2018

Accepted 29 August 2018

Available online xxxx

Keywords:

Flow shop

Batch-processing machines

Estimation of distribution algorithm

Resource consumption

ABSTRACT

The two-stage flow-shop scheduling problem investigated in this work aims minimize the resource consumption of non-identical job sizes. The flow shop consists of two batch-processing machines (BPMs): a parallel batch BPM and a sequential BPM. The makespan and resource consumption are considered together in this study, the makespan is the constraint condition, and the resource consumption is the objective. A copula-based Estimation of Distribution Algorithm (cEDA) is used to solve the problem. In this study, the individuals are coded by the allocated resource sequences of all jobs in two machines, and the convex resource consumption function is adopted to simulate the relationship between the processing time of the jobs and the resources allocated to the jobs. A Gaussian distribution is adopted as the marginal probabilistic distribution of all the components. The proposed copula function C_1 assumes independence among the components, whereas the Clayton copula function C_2 assumes that all components are interrelated and introduced for comparison. The computational experiments and comparisons verify the effectiveness of the proposed cEDA. In addition, the copula functions C_1 and C_2 adopted in the proposed cEDA approach are compared.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Resource conservation and green manufacturing have become critical in the sustainable development of modern industrial companies. A higher resource efficiency result in less expenses cost and less environmental pollution. Production systems that consider resource allocation and enhance resource efficiency, have attracted the attention of researchers in the context of manufacturing industries, such as the steel industry Müller et al. [1], Dahlström and Ekins [2], the food processing industry Henningsson et al. [3], Pagotto and Halog [4], textile industries Martínez [5], chemical industries García et al. [6] and the coating/painting industry Alkaya

and Demirel [7]. Jia et al. [8] investigated the problem of minimizing the makespan and the total electric power resource cost on a set of parallel identical batch-processing machines (BPMs). In problems that consider resource consumption, the processing time is generally determined by the amount of resources allocated.

In this work, the flow-shop scheduling problem treats resource consumption as the objective. The flow shop consists of two machines: the first machine, M_1 , is a parallel BPM, and the second machine, M_2 , is a sequential BPM. All of the jobs may have different sizes, and all of the jobs are ready to be processed at time zero. Each job must be processed successively on two machines. The processing time of each job on one machine depends on the amount of allocated resources. More resources, results in a shorter processing time for each job. The same amount of resources may result in different processing times for different jobs. On machine M_1 , all of the jobs in the same batch can be processed simultaneously, and

* Corresponding author.

E-mail addresses: chuang@mail.ustc.edu.cn (C. Liu), hpchen@ustc.edu.cn (H. Chen), rxu@hhu.edu.cn (R. Xu), rainw@mail.ustc.edu.cn (Y. Wang).

each job can share the resource allocated to the jobs belonging to the same batch. On machine M_2 , the setup time is considered. The setup operation can be performed only after the batch arrives at M_2 . The setup time (T^s) is determined by the allocated resource (u^s). The batch cannot wait between two machines (i.e., no-wait). If M_2 is busy, the batch that has been processed completely on M_1 must be blocked. In addition, the scheduling scheme must follow the rule that the constituents of each batch and the batch processing sequence on the two machines are identical. The objective of this problem is the amount of resource consumed, and the makespan C_{max} is a constraint condition that cannot be greater than the deadline D^l . If C_{max} is greater than D^l , more resources must be invested to reduce the processing time and C_{max} . Lower resource consumption typically results in higher resource utilization.

Practical applications of the problem addressed in this work can be easily found in industry. For example, Muthuswamy et al. [9] described a cleaning process observed at a sensor manufacturing facility. There is a boning operation, which establishes the electronic circuitry in a wire bonding machine. The substrates used for sensors are made of ceramic. Because residuals in the substrate could lead to defects in subsequent operations, the residuals must be removed. A substrate with residuals requires two cleaning operations in two sets of baths. In the first bath, because multiple parts that are attached to the same fixture can be cleaned at the same time, the cleaning time of a fixture is equal to the time required to clean the part with the longest cleaning time. Next, all parts belonging to the same batch move across a water jet to wash the chemicals and any additional residuals on the substrate in the second bath. Because one jet can wash only one part, a batch's cleaning time is equivalent to the sum of all of the parts in that batch. Similar practical applications can be found in [10–12]. In the manufacturing industry, optimizing resource consumption objective can reduce the cost of production, including the financial outlay, cost of manpower, cost of equipment and cost of fossil fuels. Decreasing the cost of manpower means that the manufactured products have greater competitiveness in the market, and a lower consumption of fossil fuel can result in fewer pollutants and lower waste gas purification costs [13–16].

The remainder of this study is organized as follows. Section 2 reviews the flow shop with BPMs, the resource consumption function (RCF) and the resource allocation policy of resources in a batch. We define and formulate the problem in Section 3. In Section 4, the solution representation and population initialization, the marginal probabilistic model, the proposed copula function, and the copula updating approach are explained in succession. Next, the approach for forming and sequencing batches as well as the correction algorithm for individuals are presented as well. In addition, the algorithm representation of the copula-based estimation of distribution algorithm (CEDA) is described. Computational experiments and comparisons among the different algorithms are provided in Section 5. Finally, the conclusions of this work are provided in Section 6.

2. Literature review

2.1. Flow shop with BPMs

Batch-processing problems began from the research that Lee et al. [17] initially introduced in their study of the batch-processing problem. This problem was abstracted from an industrial model of a burn-in oven in a semiconductor burn-in operation. The batch-processing operation aims to reduce the amount of resources consumed in the economic production procedure. Since then, BPMs have been widely applied in various industry sectors, ranging from traditional manufacturing to electronic chip processing. Lee et al. [17] examined the BPM of scheduling semiconductor burn-in operations. Tang et al. [18] presented a single BPM scheduling

problem considering transportation and deterioration in a steel production setting. Zhou et al. [19] considered a parallel BPM scheduling problem in the presence of non-identical job sizes and arbitrary release times.

There are two BPMs: a parallel BPM (p-BPM), which can process jobs simultaneously in the same batch and a sequential BPM (s-BPM), which can process jobs serially in the same batch. The notations p-BPM and s-BPM are adopted from Oulamara [12] to represent the type of BPM. More specially, p-BPM means that the first machine, M_1 , is a parallel BPM, and s-BPM means that the second machine, M_2 , is a sequential BPM. The aforementioned BPMs are all p-BPMs. Mathirajan and Sivakumar [20] reviewed p-BPM and s-BPM problems in the semiconductor manufacturing industry.

The studies described above seldom considered the controllable processing time of jobs with resource constraints. In addition, batch processing with a controllable processing time is more complex than normal single-machine scheduling due to the resource consumption involved. However, resource consumption, which is an economically influential factor related to the processing cost, plays an increasingly significant role in making economic and industrial decisions.

The two types of BPMs (p-BPM and s-BPM) have distinct practical applications in many fields of industrial production. The two types of BPMs have different approaches to allocating resources to jobs: p-BPM typically adopts an even resource allocation policy (ERA, such as Ng et al. [21]), whereas a s-BPM employs a manually controllable resource allocation policy (MCRA, such as Oron [22]). Batch-processing scheduling problems, which consider resource consumption, will lead to the machines' controllable processing time or setup time. In addition, the controllable processing time may increase the complexity of the problem.

There are more significant and practical multi-stage flow-shop scheduling problems that are more complex than scheduling problems with a single machine. Flow-shop scheduling problems require consideration of the flow operation from the upstream machine to the downstream machine. Multi-stage flow-shop scheduling problems are widely used in the production procedure environment.

The work in this paper can be regarded as an extension of the classical two-machines flow-shop scheduling problems with two BPMs that do not consider controllable processing time, such as [9,12,23,24]. It is easily verified that the extended problem addressed in this work can be reduced to the classical problem after affixing a resource allocation to all jobs. Once the resource allocation to all jobs is fixed, an optimal job-processing order must be determined from all possible job orders to obtain the minimum value of the makespan. In other words, the problem in this work has been converted to the classical scheduling problem, which is similar to the studies above. Those studies all addressed the two-stage flow shop, in which one machine was a parallel BPM and the other machine was a sequential BPM. However, they examined only the objective of makespan, not resource consumption. In addition, only a few researchers studied problems that consider resource consumption, and they typically treated resource consumption in the problem as the constraint, not as an objective. For example, Shabtay et al. [25] considered a no-wait two-machine flow-shop scheduling problem with convex resource-dependent processing times. Janiak et al. [26] presented a survey of resource management in machine-scheduling problems.

Some researchers have studied the resource-related objectives. For instance, Vickson [27] studied the linear combination of job-processing cost and flow cost in a single machine, where the job-processing cost was related to the amount of resources consumed. Vickson [28] analyzed a similar problem involving a tardiness criterion in two single machines. Nowicki and Zdrzałka

[29] considered both the makespan and resource consumption in a two-machine flow shop. However, the two machines were not BPMs, and the RCF was a linear function; thus, the function did not obey the law of diminishing marginal returns. Janiak [30] addressed the problem of minimizing resource consumption in the two flow-shop scheduling problem. Ng et al. [31] examined both earliness/tardiness and the amount of resources consumed in a single machine. Shabtay [32] optimized traditional scheduling objectives and considered the resource constraint.

In scheduling problems that consider resource consumption, job-processing times can be affected by the amount of resources used, such as manpower and nonrenewable fossil resources. Therefore, the processing times of jobs are not constant. The introduction of resource consumption increases the level of complexity of this type of problem Janiak et al. [26].

2.2. Resource consumption function

Traditional scheduling problems with controllable processing times generally require the RCF to be selected, because different types may not only influence the processing time of jobs and the setup time of machines, but also impact the minimization of scheduling problem objective. For example, over-simplified RCFs may not fit practical production well. In the remainder of this section, we provide a general review of the various RCFs.

Researchers typically adopt a linear resource function or a convex resource function, when examining scheduling problems in a manner that considers the resource constraint. Some researchers [31,33–35], have addressed scheduling problems with a controllable processing time, which was a linear decreasing function of the amount of resources allocated to the jobs.

Vickson [28] presented an approach to solve two single-machine sequencing problems with a processing time that was influenced by a linear RCF. Janiak [30] developed a two-processor flow-shop problem with a linear resource function, which had the form

$$p_{jv} = p_{jv}(u_{jv}) = b_{jv} - a_{jv}u_{jv}, \quad 0 \leq u_{jv} \leq \bar{u}_{jv}, \quad (1)$$

$$j = 1, 2, \dots, n, \quad v = 1, 2,$$

where a_{jv} , b_{jv} are known parameters and u_{jv} is the amount of resources allocated to job J_j in machine M_v . \bar{u}_{jv} is the upper bound on the amount of resources that can be allocated to task T_{jv} . These types of RCFs have been proposed in many papers. However, the linear RCFs have limited applicability for practical physical and economic procedures, because they fail to reflect the law of diminishing marginal returns.

To overcome that defect, some researches, [36–38], introduced a convex resource function. The convex resource function considers the law of diminishing marginal returns, an economic law that is often applied in practical production. The form of convex resource functions that are generally used in the literature is as follows:

$$p_j(r_j) = \left(\frac{w_j}{r_j} \right)^k, \quad r_j > 0, \quad k > 0, \quad (2)$$

where w_j ($w_j > 0$) is a parameter related to the workload of job J_j , and r_j ($r_j > 0$) is the amount of resources allocated to job J_j . k is a constant positive parameter that can take a value of 1 or 0.5, for example. Monma et al. [36] noted that different k values may be applicable in different problems. For example, (1) $k=1$ is applicable when the processing time of a job is inversely proportional to money or manpower resources, as in many practical government or industrial projects, and (2) $k=0.5$ corresponds to the application extracted from very large scale integration (VLSI) circuit design, where the product of silicon area resources (r_j) and the square of the time spent is a constant.

Yedidsion et al. [39] presented a general type of convex resource function that was more suitable for practical applications, and proposed four properties that the convex resource function should obey. Liang et al. [40] extended the properties proposed by Yedidsion et al. [39], and then discarded Property 4 (the job-processing time is infinite, if no resources are allocated) and proposed a new Property 5 that imposed a constraint of an initial processing time and an incompressible time for each job.

2.3. Allocation policy of resource in a batch

In a parallel BPM, all of the jobs in the same batch are processed simultaneously, and all jobs can share their entire resources allocated to each job in the same batch. The allocation approach to resources can affect the quality of the solution and thus the objective.

Three allocation policies for resources have been presented in the literature: the manually controllable resource allocation policy (MCRA, such as Oron [22], even resource allocation (ERA, such as Ng et al. [21] policy and proportional resource allocation (PRA, such as Biskup and Jahnke [41]. The essential difference between ERA and MCRA is the level of arbitrary resource allocation. MCRA can permit the resource allocation scheme, provided that the sum of the resource allocated to each job in same batch is equal to the total batch resources. However, ERA is strict in that each job in same batch receives equal resources. This means that each job belonging to the same batch shares the resources assigned to that batch in the ERA policy equally.

2.4. Copula theory

The term copula was first presented by Sklar [42]. Schweizer and Wolff [43] first studied the dependence among random variables and presented the classical Schweizer and Wolff's σ , which can measure the level of dependence among variables. In copula theory, the copula function can be viewed as a multivariate function. The formal definition of the copula function is provided in Nelsen [44]. The copula function can comprehensively reflect a multivariate relationship. Many copula functions can be used according to the definition of the copula function. Examples include the t copula function Demarta and McNeil [45], the Clayton copula and the Archimedean copula Whelan [46].

In addition, Nelsen [44] proved Sklar's Theorem in n dimensions, which is the essential to the theory of the copula-based estimation of distribution algorithm (cEDA). The n -dimensional Sklar's theorem can be described as follows Nelsen [44]:

Theorem 2.1. *Let H be an n -dimensional distribution function with margins F_1, F_2, \dots, F_n . Then there exists an n -copula C such that for all \mathbf{x} in $\bar{\mathbf{R}}^n$ ($\bar{\mathbf{R}}$ is the closed set $[-\infty, +\infty]$),*

$$H(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n)) \quad (3)$$

If F_1, F_2, \dots, F_n are all continuous, then C is unique; otherwise, C is uniquely determined on $\text{Ran}F_1 \times \text{Ran}F_1 \times \dots \times \text{Ran}F_n$. ($\text{Ran}F_n$ denotes the Range of F_n)

According to Theorem 2.1, a multivariate joint distribution can be divided into two parts, each random variable marginal distribution function and a copula function. Therefore, each random variable marginal distribution and the proposed copula function to be used in this work can be determined separately. Introducing the copula function to EDAs can simplify the process of building the probability distribution model and reduce its performing time. A more detailed description of copula theory can be found in Nelsen [44].

Table 1
Notations.

Notation	Description
n	number of jobs
j	index of the job, $j = 1, 2, \dots, n$
v	index of the machine, $v = 1, 2$
i	index of the batch, $i = 1, 2, \dots, m$
m	number of batches
J_j	j th job
S_j	size of j th job
T_{jv}	task of job J_j on machine M_v
w_{jv}	workload of task T_{jv}
M_v	v th machine
B_i	i th batch
u_{jv}	resource allocation variable for task T_{jv}
\bar{u}	maximal amount of resources allocated to all tasks
\mathcal{R}	resource allocation vector of all tasks, $\mathcal{R} = (u_{11}, u_{21}, \dots, u_{n1}; u_{12}, u_{22}, \dots, u_{n2})$
r_j	amount of resources owned by the j th job, $r_j = u_{j1} + u_{j2}$
U_{iv}	total resources allocated to batch B_i on machine M_v
D^l	common deadline for all jobs
$p_{jv}(u_{jv})$	processing time of task T_{jv} with the u_{jv} resource
$e_{jv}(u_{jv})$	resource utility of J_j on machine M_v
P_{iv}	processing time of batch B_i on machine M_v
T^s	setup time of machine M_2
u^s	amount of resource allocated to each batch for setup on machine M_2
Θ	batch capacity on machine M_1
Π	job sequence of all jobs
X_{ji}	1, if job J_j belongs to batch B_i ; 0, otherwise
Y_i	1, if batch B_i is not empty; 0, otherwise
Ψ_{i1}	starting time of batch B_i processed on machine M_1
C_{i1}	department time of batch B_i processed on machine M_1
C_{i2}	complete time of batch B_i processed on machine M_2
C_{max}	makespan

3. Problem formulation

A resource allocation vector of all jobs on two machines and a job processing order are required to be found, such that the total resource consumption is minimized under the constraint of a given deadline D^l . For a given resource allocation vector (\mathcal{R}), there are many available job orders to be selected, but a different job-processing order will yield a different C_{max} . If the C_{max} is larger than deadline D^l , that resource allocation vector (\mathcal{R}) will be not valid, and should be amended by the correction algorithm.

The problem is formulated in this section. To formulate the problem to be addressed, we will first provide all of the notations employed in this work and then formulate the objective, constraints, decision variables and proposed RCF function. Next, the mathematical formulation of the addressed problem can be obtained.

3.1. Notations

The notations used in this work, such as the subscripts and variables, are summarized in Table 1.

3.2. Hypotheses of the problem

To clearly describe the problem to be addressed in this work, the following hypotheses will be stated in detail:

1. There are n independent and non-identical jobs. J_j denotes the j th ($j = 1, 2, \dots, n$) job, and s_j denotes the size of job J_j . For job J_j , the processing procedure thus consists of two tasks, T_{j1} on M_1 and T_{j2} on M_2 .
2. The allocation unit of the resource is task T_{jv} ; the amount of resources assigned to task T_{jv} is u_{jv} . The processing time of task T_{jv} on machine M_v is $p_{jv}(u_{jv})$. Therefore, the total amount of resources assigned to job J_j is r_j , which is the sum of u_{j1} and u_{j2} .

3. The capacity of the first machine, M_1 , is denoted by Θ , and machine M_1 can process a batch of jobs provided that the total size of all jobs assigned to the batch does not exceed its capacity. Once the job is processed in the machine, it cannot be preempted. After finishing processing the batch on machine M_1 , that batch can either block machine M_1 or go to machine M_2 directly for processing, based on whether machine M_2 is busy.
4. The processing time of batch B_i on machine M_1 is equal to the longest job-processing time among all of the jobs in that batch. The processing time of batch B_i on M_1 can be calculated by $P_{i1} = \max\{p_{j1} | j \in B_i\}$. On machine M_2 , the processing time of batch B_i is equivalent to the sum of the processing times of all jobs in the same batch.
5. The ERA policy is adopted on machine M_1 , and the MCRA policy is adopted on machine M_2 in this work.
6. The objective of the problem in this work is the total amount of the resource consumed in the scheduling process.
7. The amount of resources allocated to each batch on machine M_2 for the setup operation is the constant u^s , and the corresponding time is T^s .
8. The makespan C_{max} should not exceed the given deadline D^l .

3.3. Proposed resource consumption function

Considering the law of diminishing marginal returns in the production and processing environment, the RCF adopted in this work can be expressed as follows Yedidsion et al. [39]:

$$p_{jv}(u_{jv}) = \left(\frac{w_{jv}}{u_{jv}} \right)^k, u_{jv} > 0 \quad (4)$$

where w_{jv} is the parameter related to the workload of job J_j on machine M_v , and u_{jv} is the resources allocated to job J_j on machine M_v . The proposed RCF obeys all four properties from Yedidsion et al. [39].

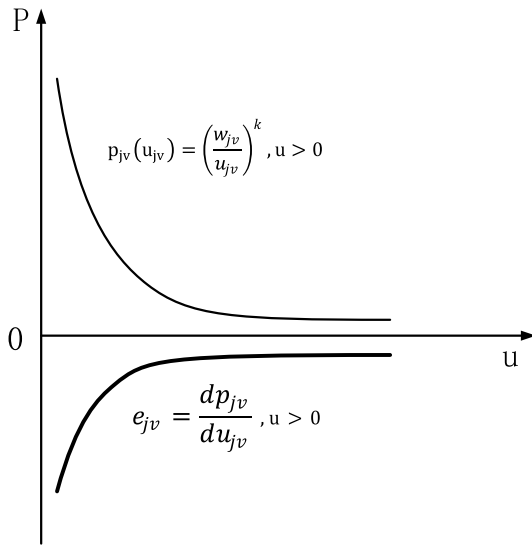


Fig. 1. RCF and resource sensitivity.

The concept of resource sensitivity from Liang et al. [40], which is the derivative of the processing time $p_{jv}(u_{jv})$ on resource (u_{jv}), is introduced as

$$e_{jv} = \frac{dp_{jv}(u_{jv})}{du_{jv}}, u_{jv} > 0 \quad (5)$$

where e_{jv} denotes the derivative of $p_{jv}(u_{jv})$ on u_{jv} .

The curves of the RCF ($p_{jv}(u_{jv})$) and resource sensitivity (e_{jv}) are illustrated in Fig. 1. A larger value of u_{jv} , is associated with less diminishing marginal returns.

3.4. Mathematical model

According to the 3-field problem classification scheme, $\alpha|\beta|\gamma$, for scheduling problems Graham et al. [47], the problem in this work can be described as follows:

$$F_2|p - BPM, s - BPM, no - wait, s_j, p_{jv}(u_{jv}) \\ = (\frac{w_{jv}}{u_{jv}})^k, C_{max} \leq D^l | \sum_{j=1}^n r_j + mu^s, \quad (6)$$

where D^l is the deadline assigned by the scheduler. Based on the introduction of the parameters and the detailed description of the problem set forth above, the mathematical model of the problem in this work is formulated as follows:

$$\text{Minimize} \quad \sum_{j=1}^n r_j + m \cdot u^s \quad (7)$$

Subject to:

$$\sum_{i=1}^m X_{ji} = 1, \quad j = 1, \dots, n \quad (8)$$

$$\sum_{j=1}^n s_j X_{ji} \leq \Theta, \quad i = 1, \dots, m \quad (9)$$

$$C_{max} \leq D^l, \quad i = 1, \dots, m \quad (10)$$

$$\Psi_{i1} = C_{i-1,1}, \quad i = 2, \dots, m \quad (11)$$

$$C_{i1} \geq \Psi_{i1} + P_{i1}, \quad i = 1, \dots, m \quad (12)$$

$$C_{i2} \geq C_{i1} + P_{i2} + T^s, \quad i = 1, \dots, m \quad (13)$$

$$C_{max} \geq C_{i2} \geq C_{i1} \geq \Psi_{i1} \geq 0, \quad i = 1, \dots, m \quad (14)$$

Objective (7) minimizes the amount of resource consumed. Constraint (8) ensures that each job is assigned to only one batch. Constraint (9) ensures that the total size of jobs in the same batch does not exceed the machine's capacity. Constraint (10) ensures that makespan C_{max} is not greater than deadline D^l . Constraint (11) requires that the starting time of batch B_i on machine M_1 is equal to the department time of batch B_{i-1} on machine M_1 . Constraint (12) indicates that the department time of the batch B_i on machine M_1 is greater than the sum of starting time and its processing time on machine M_1 . Constraint (13) determines the completion time of each batch on machine M_2 . Constraint (14) determines the makespan C_{max} .

Because $r_j = u_{j1} + u_{j2}$ and $U_i = \sum_{j \in B_i} u_{jv}$, the objective in Eq. (7) can be deduced as the following formulations:

$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^n r_j + m \cdot u^s \\ = \text{Min} \quad & \sum_{j=1}^n u_{j1} + \sum_{j=1}^n u_{j2} + m \cdot u^s \\ = \text{Min} \quad & \sum_{i=1}^m U_{i1} + \sum_{i=1}^m U_{i2} + m \cdot u^s \end{aligned} \quad (15)$$

4. Proposed cEDA

In reality, the cEDA is a novel improved Estimation of Distribution Algorithm (EDA) based on the copula function for addressing optimization problems.

EDA approaches can be divided into two categories, the discrete EDA for discrete optimization problems, and the continuous EDAs for continuous optimization problems. Ceberio et al. [48] gave a review on EDAs in permutation-based combinatorial optimization problems. On the one hand, discrete EDAs can be classified as univariate, bivariate and multivariate according to the probabilistic model employed. Univariate EDAs assume that all of the variables are independent; examples include the PBIL Baluja [49], Sebag and Ducoulombier [50] and the UMDA Larrañaga [51]. Bivariate EDAs consider dependencies between pairs of variables; examples include the BMMA Pelikan and Muehlenbein [52] and the MIMIC De Bonet et al. [53]. Multivariate EDAs consider multivariate correlations, such as the Bayesian optimization algorithm Pelikan et al. [54]. And Ceberio et al. [55] proposed a distance-based ranking model EDA for the flowshop scheduling problem. On the other hand, the continuous EDAs are developed on the basis of discrete EDA, the idea of many algorithms from discrete EDA. The UMDAC approach Larrañaga et al. [56] is extended from the UMDA approach above. The PBILC approach Sebag and Ducoulombier [50] is developed from the PBIL approach above. The MIMIC_c approach Larrañaga and Lozano [57] is derived from the MIMIC approach above.

cEDAs can address the marginal distribution and the correlation among all components separately. Wang et al. [58] proved that cEDAs can simplify the parameter learning operation and sampling procedure by incorporating the copula function into the EDA. The related studies that consider EDAs based on the copula function are described here. Ye et al. [59] proposed a cEDA based on a class of nested Archimedean copulas constructed with Lery subordinators. Cuesta-Infante et al. [60] investigated the use of empirical and Archimedean copulas in EDAs. De Mello et al. [61] proposed a cEDA based on a multivariate extension of the Archimedean with parameter updating for number optimization problems. Salinas-Gutiérrez et al. [62] presented the use of copula functions and graphical models in EDAs. Gao et al. [63] considered multi-objective optimization problems by incorporating the multivariate Archimedean copulas function into EDAs.

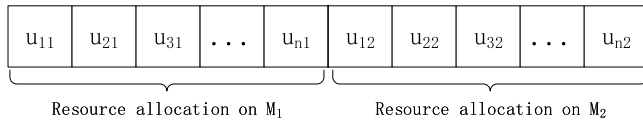


Fig. 2. Illustration of the solution representation.

Considering the complexity of the correlations among all of the components in this problem, and the fact that few studies treat resource consumption as the objective, the cEDA approach is employed in this study to fill this gap in the literature. The key cEDA procedures Wang et al. [58] can be described as follows:

- Randomly initialize the population.
- Calculate the fitness value of all individuals.
- Select the better individuals according to the fitness function and selection policy.
- Construct the copula distribution model.
- Sample new individuals from the constructed distribution model.

In this section, the solution representation and population initialization will be introduced first. The proposed marginal probabilistic model and two different copula functions used in this work can be illustrated successively. In addition, to calculate the makespan C_{max} of each scheduling scheme, this section describes the proposed approach for forming and sequencing batches and the correction algorithm for individuals. Finally, the detailed algorithm representation of cEDA is given.

4.1. Solution representation and population initialization

The solution to this problem can be described as follows. A solution is a resource vector allocated to each job in two machines (M_1 and M_2). The amount of the component in a resource vector is equal to $2n$, which is the product of the number of machines ($v = 2$) and the number of jobs (n). In other words, the solution can be denoted as $(u_{11}, u_{21}, \dots, u_{n1}; u_{12}, u_{22}, \dots, u_{n2})$, where the top n components are the resources allocated to the n jobs in machine M_1 and the last n components are the resources allocated to the n jobs in machine M_2 . The representation of the solution is illustrated in Fig. 2.

The initial population can impact the convergence speed of the cEDA and the solution quality of the problem. Considering the diversity of the initial population, the initial population (N individuals) is generated randomly from a uniform distribution in the solution space.

4.2. Marginal distribution function and copula function

The marginal distribution functions represent the distribution of the components, and the copula function represents the correlation among all components. In the literature related to probabilistic models of EDA, the Gaussian distribution is easily found, such as Lu and Yao [64], Yuan and Gallagher [65], Wang et al. [58] and Su and Chow [66].

The Gaussian distribution is adopted as the marginal distribution function for all components u_{jv} ($j = 1, 2, \dots, n$, $v = 1, 2$). For two proposed distinct copula functions, the proposed marginal distribution functions are all Gaussian distributions. In particular, each component in the resource vector $(u_{11}, u_{21}, \dots, u_{n1}; u_{12}, u_{22}, \dots, u_{n2})$ obeys the Gaussian distribution $N(\mu_{jv}, \sigma_{jv}^2)$. For component u_{jv} , its marginal distribution function $F_{jv}(u_{jv})$ can be formulated as follows:

$$F_{jv}(u_{jv}) = \int_{-\infty}^{u_{jv}} \frac{1}{\sqrt{2\pi}\sigma_{jv}} \exp\left(-\frac{(x - \mu_{jv})^2}{2\sigma_{jv}^2}\right) dx \quad (16)$$

For resource vector $(u_{11}, u_{21}, \dots, u_{n1}; u_{12}, u_{22}, \dots, u_{n2})$ in this work, the interrelationship among $2n$ components is considered independent, and the non-independent relationship among all components is adopted as a comparison.

Theorem 4.1. Let x_1, x_2, \dots, x_n be continuous random variables. Then x_1, x_2, \dots, x_n are independent if and only if $C_{x_1, x_2, \dots, x_n} = \prod(F_1(x_1), \dots, F_n(x_n))$.

According to Theorem 4.1 from Nelsen [44] set forth above, the proposed copula function C_1 , which assumes that all variances are independent, can be described as follows:

$$C_1 = C_{u_{11}, \dots, u_{n2}} = \prod(F_{11}(u_{11}), \dots, F_{n1}(u_{n1}), F_{12}(u_{12}), \dots, F_{n2}(u_{n2})) \quad (17)$$

From Eq. (17), copula function C_1 is the product of the independent univariate marginal distribution functions.

To compare the effectiveness of different copula functions in this scheduling problem, another copula function C_2 , in which all components are non-independent, is introduced. Copula function C_2 is a nested Clayton copula function that is the product of two Clayton copula functions. In copula function C_2 , the interrelation between the components in machines M_1 and M_2 is independent. Function C_2 can be described as follows:

$$C_2 = \prod(C_a, C_b), \quad (18)$$

where C_a and C_b are both Clayton copula functions. Copula functions C_a and C_b handle the resource components in machines M_1 and M_2 , respectively. The Clayton copula functions C_a and C_b that are introduced from Schweizer and Sklar [67] are formulated in Eq. (19). The generator φ for Clayton copula functions C_a and C_b are $\varphi(t) = t^{-\theta} - 1$ ($\theta \geq 0$).

$$\begin{cases} C_a = \left(1 + \sum_{j=1}^n (u_{jv}^{-\theta_a} - 1)\right)^{-\frac{1}{\theta_a}}, & v = 1, \\ C_b = \left(1 + \sum_{j=1}^n (u_{jv}^{-\theta_b} - 1)\right)^{-\frac{1}{\theta_b}}, & v = 2. \end{cases} \quad (19)$$

The definition and related details of Clayton copula function C_a and C_b are provided in Alsina et al. [68], Nelsen [44].

Next, copula function C_2 in Eq. (18) can be deduced as follows:

$$C_2 = \prod(C_a, C_b) = \left(1 + \sum_{j=1}^n (u_{j1}^{-\theta_a} - 1)\right)^{-\frac{1}{\theta_a}} \cdot \left(1 + \sum_{j=1}^n (u_{j2}^{-\theta_b} - 1)\right)^{-\frac{1}{\theta_b}}, \quad (20)$$

Marshall and Olkin [69] proposed a sampling approach for copula functions C_a and C_b , and the details of the sampling approach for the Clayton copula function are provided in Section 4.3. The cEDA approach with copula function C_1 is denoted by $cEDA_{c1}$, and the cEDA approach with copula function C_2 is denoted by $cEDA_{c2}$. In the remaining sections, $cEDA_{c1}$ is abbreviated as cEDA, if not explicitly specified.

4.3. Copula updating approach: learning and sampling

As noted above, the cEDA approach generates new individuals by sampling from the copula distribution model, potentially influencing the performance of the proposed cEDA approach. This section discusses the learning of the parameters of the copula distribution model and sampling from the constructed copula distribution model.

Table 2
Details of S individuals.

u_{11}	u_{21}	\dots	u_{n1}	u_{12}	u_{22}	\dots	u_{n2}
u_{11}^1	u_{21}^1	\dots	u_{n1}^1	u_{12}^1	u_{22}^1	\dots	u_{n2}^1
u_{11}^2	u_{21}^2	\dots	u_{n1}^2	u_{12}^2	u_{22}^2	\dots	u_{n2}^2
u_{11}^3	u_{21}^3	\dots	u_{n1}^3	u_{12}^3	u_{22}^3	\dots	u_{n2}^3
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_{11}^S	u_{21}^S	\dots	u_{n1}^S	u_{12}^S	u_{22}^S	\dots	u_{n2}^S

Note: u_{jv}^k is u_{jv} of the k th individual.

The truncation selection policy is employed to select S individuals with the best objective values out of N individuals in each generation. The fitness function adopted to evaluate all of the individuals can be formulated as follows:

$$\sum_{j=1}^n r_j + m \cdot u^s, \quad (21)$$

where r_j is the amount of the resources allocated to job J_j , m is the number of batches formed in the algorithm, and u^s is the amount of resources consumed for a constant setup time T^s in machine M_2 .

S selected individuals in the current generation can be used to determine the $2n$ parameters $(\mu_{11}, \sigma_{11}; \dots; \mu_{n2}, \sigma_{n2})$. The details of S individuals (samples) are summarized in Table 2.

The parameters $(\mu_{jv}$ and $\sigma_{jv}^2)$ of all marginal distributions can be estimated individually in the cEDA approach. More specially, the first column (u_{11}) in Table 2, which consists of S components $(u_{11}^1, u_{11}^2, \dots, u_{11}^S)$ from the different S individuals, can be used to estimate the parameters μ_{11} and σ_{11}^2 using the Maximum Likelihood Estimate (MLE) approach; thus, the other columns are the same.

For component u_{jv} in the resource vector, there are S values $\{u_{jv}^1, u_{jv}^2, \dots, u_{jv}^S\}$ from S selected individuals, which can be used to learn the parameters $(\mu_{jv}$ and $\sigma_{jv}^2)$ of the marginal distribution of component u_{jv} . To simplify the equation, $u_{jv}^i (i = 1, 2, \dots, S)$ is denoted by x_i . The following equation can be obtained according to the MLE approach:

$$\begin{aligned} L(\mu_{jv}, \sigma_{jv}^2) &= \prod_{i=1}^S \frac{1}{\sqrt{2\pi}\sigma_{jv}} \exp\left(-\frac{(x_i - \mu_{jv})^2}{2\sigma_{jv}^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma_{jv}}\right)^S \exp\left(-\sum_{i=1}^S \frac{(x_i - \mu_{jv})^2}{2\sigma_{jv}^2}\right) \end{aligned} \quad (22)$$

where μ_{jv} and σ_{jv}^2 are the mean and variance of the distribution function of component u_{jv} , respectively, and the results can be derived as follows:

$$\begin{cases} \hat{\mu}_{jv} = \frac{1}{S} \sum_{i=1}^S x_i = \bar{x} \\ \hat{\sigma}_{jv}^2 = \frac{1}{S} \sum_{i=1}^S (x_i - \bar{x})^2 \end{cases} \quad (23)$$

Through the analysis above, the procedure of learning the parameters $(\mu_{jv}$ and $\sigma_{jv}^2)$ using the MLE approach for the problem in this study can be summarized as follows:

1. Prepare the data $\{u_{jv}^1, u_{jv}^2, \dots, u_{jv}^S\}$.
2. Calculate $\hat{\mu}_{jv} = \bar{x} = \frac{1}{S} \sum_{i=1}^S u_{jv}^i$.
3. Calculate $\hat{\sigma}_{jv}^2 = \frac{1}{S} \sum_{i=1}^S (u_{jv}^i - \bar{x})^2$.

After learning the parameters of all marginal distributions, the copula distribution model can be obtained according to the copula

function in Eq. (17). Next, the copula function can be used to generate new individuals, i.e., the point $(x_{11}, x_{21}, \dots, x_{n1}; x_{12}, x_{22}, \dots, x_{n2})$ in $2n$ dimensions, can be gained. Then, a new individual $(u_{11}^{new}, u_{21}^{new}, \dots, u_{n1}^{new}; u_{12}^{new}, u_{22}^{new}, \dots, u_{n2}^{new})$ can be obtained according to the equation $u_{jv}^{new} = F_{jv}^{-1}(x_{jv})$. The generated new individuals belong to the next generation.

The individuals of the next-generation population consist of 3 parts:

1. M better individuals (elites), which have the best objective values from the current generation.
2. L new individuals formed by a uniform distribution in the solution space;
3. $N - M - L$ new individuals sampled from the copula function.

In this work, the S individuals selected to construct the copula probability model are only partly reserved for the next generation, which means that $M \leq S$. The number M of the best individuals can be selected from the most recent generation according to the fitness function. To increase the diversity of the population, L new individuals are sampled randomly from a uniform distribution in the solution space. The remaining $N - M - L$ new individuals are sampled from the copula function, which can guarantee the global exploration ability of the proposed cEDA approach.

More specifically, the sampling approach Marshall and Olkin [69] for the Clayton copula function C_a is illustrated in Algorithm 1. θ_a is a parameter, and $\hat{\mu}_{jv}$ and $\hat{\sigma}_{jv}$ are the parameters obtained from the marginal distribution function of component u_{jv} . Function $F(v)$ in Algorithm 1 is the product of two Gamma $\Gamma(\alpha, \beta)$ distribution functions, in which α is equal to $1/\theta$ and β is equal to 1, i.e., $v \sim \Gamma(\frac{1}{\theta}, 1)$. The sampling approach for the Clayton copula function C_b is the same as that for C_a .

The difference between copula functions C_1 and C_2 is the manner in which the dependencies among all of the components are handled. In C_1 , all of the components are independent. In C_2 , the correlation among all of the components are non-independent and complex, and the components $(u_{11}, u_{21}, \dots, u_{n1})$ and $(u_{12}, u_{22}, \dots, u_{n2})$ are independent. However, the correlation among $u_{11}, u_{21}, \dots, u_{n1}$ is non-independent, and this complex correlation is addressed by Clayton copula function C_a . The non-independent correlation among $u_{12}, u_{22}, \dots, u_{n2}$ is addressed by Clayton copula function C_b . Therefore, the sampling operation of copula function C_2 can be achieved by the sampling operation of copula function C_a and C_b .

Algorithm 1 Sampling Algorithm for the Clayton copula function

- 1: **for** $k = 1 \rightarrow N - M - L$ **do**
- 2: Randomly generate v according to the distribution function;
 $F(v) = \frac{1}{\Gamma(1/\theta_a)} e^{-v} \cdot v^{1/\theta_a - 1}$;
- 3: **for** $i = 1 \rightarrow n$ **do**
- 4: Randomly generate independent v_i from the uniform distribution of interval $[0, 1]$;
- 5: $x_i = \Phi^{-1}((- \ln v_i)/v) = (1 - \frac{\theta_a \ln v_i}{v})^{-1/\theta_a}$;
- 6: **end for**
- 7: Calculate $u_i^k = F_i^{-1}(x_i; \hat{\mu}_i, \hat{\sigma}_i)$;
- 8: **end for**

4.4. Forming batches and batches sequencing decision

The next fit (NF) approach, which is adopted for the bin-packing problem, can be applied to BPMs because of the similarity between bin packing and batch generating. The NF approach employed by Muthuswamy et al. [9] is used to generate batches in this study.

The jobs sorting operation and the general procedure of the NF approach are described in Algorithm 2.

Algorithm 2 Next Fit Algorithm

```

1: Arrange jobs in some random order;
2:  $i \leftarrow 1$ ;
3:  $B_i \leftarrow$  Create an initial batch;
4:  $ResidualCap \leftarrow$  Define the residual capacity of batch  $B_i$ ;
5:  $ResidualCap \leftarrow \Theta$ ;
6: for  $j = 0 \rightarrow n$  do
7:   if  $s_j \leq ResidualCap$  then
8:     Pack the job  $J_j$  to batch  $B_i$ ;
9:      $ResidualCap \leftarrow ResidualCap - s_j$ ;
10:  else
11:     $i \leftarrow i + 1$ ;
12:    Create a new batch  $B_i$ ;
13:  end if
14: end for

```

After the operation of forming batches is complete, each job belongs to only one batch. Next, based on the approach from Muthuswamy et al. [9] and the specific character in our problem, we develop a heuristic, called the Least machine Idle Time Heuristic (LITH), for sequencing the formed batches. The fundamental idea is as follows:

1. Randomly determine the first batch to be processed.
2. Calculate the idle time of all residual unscheduled batches.
3. Choose the next batch that can result in the least idle time on two machines.
4. Repeat the three procedures above until all batches are scheduled.

The pseudo code of the LITH algorithm is described in Algorithm 3; C_{kv} and C_{iv} in Algorithm 3 represent the complete time of batches B_k and B_i , respectively, on machine M_v , and the total idle time of the two machines is denoted by Q_i .

Algorithm 3 LITH Algorithm

```

1: Sequence all batches randomly ;
2: Rename all batches by their sequence index:  $B_1, B_2, B_3, \dots$ ;
3: Determine the head( $B_1$ ) in the batch sequence as the batch to be processed first;
4: Calculate  $C_{11}$  and  $C_{12}$  of batch  $B_1$ ;
5: Delete batch  $B_1$  from the batch sequence;
6: for  $k = 1 \rightarrow m - 2$  do
7:   for  $i = 1 \rightarrow m - k$  do
8:      $C_{i1} \leftarrow \max\{C_{k1} + P_i, C_{k2}\}$ ;
9:      $Q_i \leftarrow (C_{i1} - P_i - C_{k1}) + (C_{i1} - C_{k2})$ ;
10:  end for
11:  Select batch  $B_i$  with the lowest value of  $Q_i$ ;
12:  Delete batch  $B_i$  from the batch sequence;
13:  Calculate  $C_{k+1,1}$  and  $C_{k+1,2}$ ;
14: end for

```

Fig. 3 provides an example illustrating the LITH approach. The related batch-processing times on the two machines are displayed in Table 3. The shadow parts in Fig. 3 are the idle time on the two machines. The first batch to be processed is assumed to be batch B_1 . Then, the next batch to be processed must be chosen from the batch set $\{B_2, B_3, B_4\}$. According to Fig. 3, the idle time of schedule S1 is 2, the idle time of schedule S2 is 1, and the idle time of schedule S3 is 2. Therefore, batch B_3 is selected to be processed after batch B_1 , according to the LITH algorithm.

Table 3

Batch-processing time on the two machines.

Batch	Batch-processing time on M_1	Batch-processing time on M_2 ^a
B_1	3	4
B_2	2	3
B_3	3	5
B_4	6	4

^aNote: All batch-processing times contain the setup time on M_2 .

4.5. Correction algorithm for individuals

Some of the initial individuals in each generation may not obey the deadline (D^l) constraint. Therefore, these individuals must be amended. Based on the correction procedure from Liang et al. [70], a new advanced correction algorithm is proposed to amend the individuals that disobey the scheduling constraint $C_{max} \leq D^l$. This advanced correction algorithm can dynamically adjust the component to be amended in an individual, according to the change of e_{jv} . The correction algorithm must be executed before the selection operation in each generation. The procedure for the correction algorithm is described in Algorithm 4.

Algorithm 4 Correction Algorithm

```

1: Calculate  $|e_{jv}|$  of all components in one individual;
2: while  $C_{max} > D^l$  do
3:   Determine the maximum value of  $|e_{jv}|$  among all components;
4:   Task  $T_{jv} \leftarrow$  Select the task corresponding to the maximum  $|e_{jv}|$ ;
5:   Add a certain amount of resources to the task  $T_{jv}$ ;
6:   Update  $e_{jv}$  of task  $T_{jv}$ ;
7:   Calculate  $C_{max}$ ;
8: end while

```

4.6. Algorithm representation of cEDA

Based on the details set forth above, the pseudo code of the cEDA algorithm can be presented in Algorithm 5. In the remainder of this subsection, the computational cost of cEDA is analyzed following the steps in Algorithm 5.

The initial solutions are generated at random with the time complexity of $O(Nn)$, where the N is the population size and the n is the number of jobs. For calculating the makespan of each individual, the forming batches and batches sequencing operation need to be done. The forming batches costs $O(n)$ time, and the batches sequencing operation takes $O(m \log m)$ time, where m is the number of batches. Thus, for the population which has N individuals, the time complexity is $O(N(n + m \log m))$. For the individual which violates the deadline constraint, the time complexity of the correction procedures is $O(nk)$, where the k is the number of iterations of the loop in Algorithm 4. In the selection operation, all individuals are sorted by objective value. So the time complexity is $O(N \log N)$. In the copula learning process, $2n$ pairs of μ_{jv} and σ_{jv}^2 need to learn. The learning operation has a close complexity of $O(nS)$, where S is the number of better individuals selected from the population. The sampling operation from uniform distribution has a complexity of $O(Ln)$, where the L is the number of new generating individuals. In the sampling procedure from copula joint probability model, generating new individuals can be done in $O(Nn)$ time. And confirming that whether each individual satisfies the deadline constraint takes $O(n + m \log m)$ time.

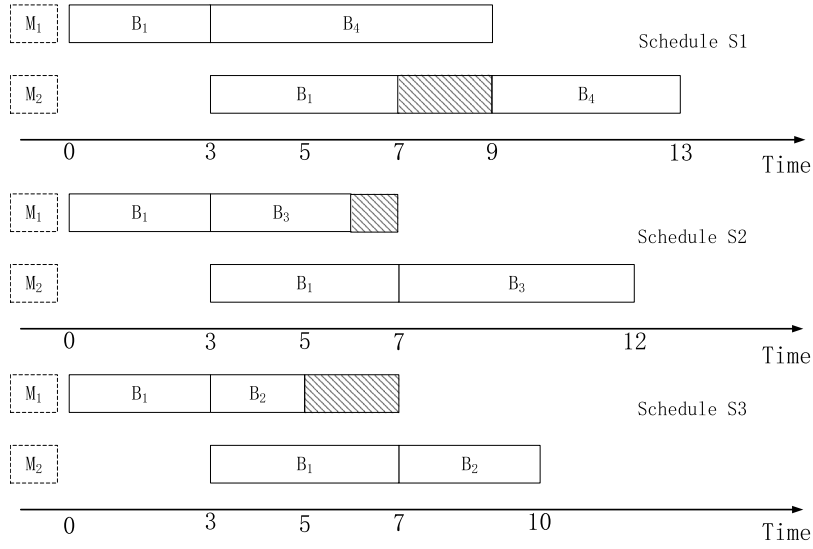


Fig. 3. Example for the LITH approach.

Algorithm 5 General framework of cEDA

```

1: Initialize parameters  $p_1$  and  $p_2$ ;
2:  $S \leftarrow p_1 \cdot N$ ;  $M \leftarrow p_2 \cdot N$ ;  $L \leftarrow 5\% \cdot N$ ;
3:  $0 \leftarrow$  Generation counter of population  $l$ ;
4:  $l_{max} \leftarrow$  Maximum value of generation;
5:  $G_0 \leftarrow$  Generate  $N$  initial individuals, according to the uniform distribution in the solution space;
6: while Stopping condition is not met do
7:   for  $k = 1 \rightarrow N$  do
8:     Calculate the  $C_{max}$  of the  $k$ th individual in the current population;
9:     if  $C_{max} > D^l$  then
10:      Use the correction algorithm to amend the  $k$ th individual;
11:     end if
12:   end for
13:    $G_l^{better} \leftarrow$  Select  $S$  better individuals from  $G_l$  according to the selection policy;
14:   for  $k = 1 \rightarrow 2n$  do
15:     Learn the  $k$ th marginal probability distribution of  $G_l^{better}$ ;
16:   end for
17:   Build the copula joint probability model;
18:   Sample  $L$  new individuals from the uniform distribution in the solution space;
19:   Sample  $N - M - L$  new individuals from the copula joint probability model;
20:    $G_{l+1} \leftarrow$  Merge  $G_l^{better}$  with the individuals sampled from Line 18 and Line 19;
21:    $l \leftarrow l + 1$ ;
22: end while

```

5. Computational experiments**5.1. Experimental design**

The experiment is divided into four parts. First, $cEDA_{c1}$ and $cEDA_{c2}$ are compared. Second, the genetic algorithm (GA), which is a classical evolutionary algorithm, is introduced to compare with the proposed $cEDA_{c1}$ approach. Third, the particle swarm optimization (PSO) approach, which is a computational swarm intelligence approach, is adopted for comparison with the $cEDA_{c1}$ approach.

Table 4

Parameter settings of the experiment.

Parameter	Value
Number of jobs (n)	10, 20, 50, 100 and 200
Size of the jobs	Uniform[1, 5] and Uniform[1, 15]
Workload of the jobs	Uniform[1, 8] and Uniform[1, 16]
Setup time (T^s) of machine M_2	2
Resource consumption for setup	3
Capacity of the first machine M_1 (Θ)	20
Common deadline (D^l) for all jobs	$2n$ and $2n - 5$

Fourth, an EDA approach, named $MIMIC_c$ algorithm, is adopted and compared for demonstrating the advantages of the proposed $cEDA_{c1}$. Finally, the convergence trends of the four algorithms are also given in this section. In this section, $cEDA_{c1}$ is abbreviated as cEDA, if not explicitly specified. All of the algorithms are executed in MATLAB R2017a(9.0.0.341360) using a PC(Win10, 64-bit) with an Intel(R) Core(TM)i7-6700 3.4 GHz CPU and 16.00 GB RAM.

The initial population including N individuals, is generated randomly from the uniform distribution in the solution space. The problem instances are generated by following the approach of Muthuswamy et al. [9], and the parameters settings employed for cEDA are determined: the number of jobs, the size of the jobs, the workload of jobs, the setup time T^s on machine M_2 , the resource consumed for setup and the capacity of the first machine. The parameter settings are presented in Table 4. For each number of jobs, 8 configurations that consider different job sizes (small and large), workloads (low and high) and deadlines (short and long) are taken into account. For the small-scale instances (10 and 20 jobs), medium-scale instances (50 and 100 jobs) and large-scale instances (200 jobs), 3 instances are generated in each configuration. Each algorithm is executed 10 times for each instance.

The value of k in Eq. (4) is equal to $\frac{1}{2}$. Next, the proposed RCF can be presented as follows:

$$p_{jv}(u_{jv}) = \left(\frac{w_{jv}}{u_{jv}} \right)^k = \sqrt{\frac{w_{jv}}{u_{jv}}}, u_{jv} > 0, \quad (24)$$

where s_j is the size of job J_j . e_{jv} can be deduced from Eq. (24) as follows:

$$e_{jv}(u_{jv}) = -\frac{1}{2} \sqrt{\frac{w_{jv}}{u_{jv}^3}}, u_{jv} > 0, \quad (25)$$

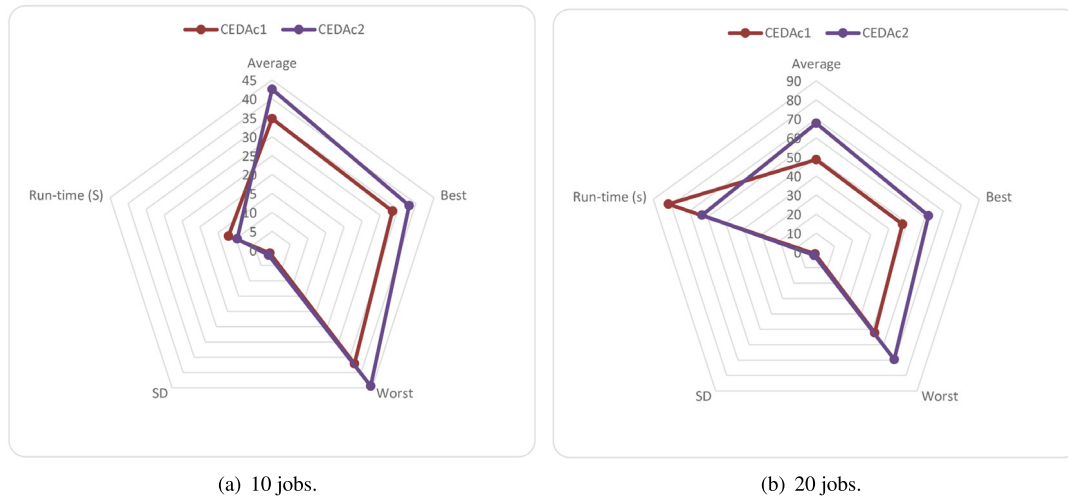


Fig. 4. $cEDA_{c1}$ vs. $cEDA_{c2}$, the size of jobs obeys $U[1,5]$, the workload obeys $U[1,8]$.

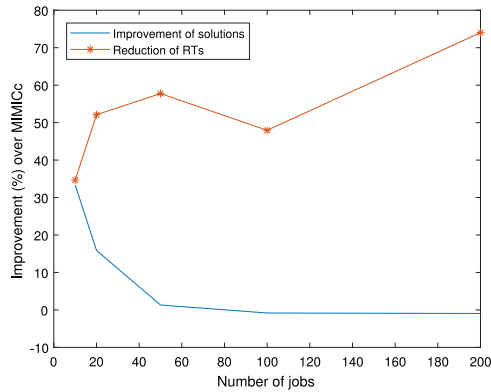


Fig. 5. Comparison of improvement of solution and reduction of running time under different number of jobs. The definition of improvement of solutions and reduction of RTs can be found in Eqs. (33) and (34), respectively.

5.2. Parameter settings

Unless stated specifically, the default settings of the $cEDA$ approach used in the following experiments are as follows:

1. The population size is set to $N = 6n$.
2. The truncation selection policy is adopted, and the selection ratio is set to p_1 , which means that the best S ($S = p_1 \cdot N$) individuals in the current generation can be selected to learn the parameters of the marginal distribution function. The value of p_1 can be determined by preliminary experiments in this section.
3. $M = p_2 \cdot N$. The value of p_2 can be determined through a preliminary experiment in this section.
4. L ($L = 5\%N$) individuals can be sampled from the uniform distribution in the solution space.
5. The stopping criterion is set to be the maximum generation. For the configuration of n jobs, the maximum generation is set to $18n$. Meanwhile, when the best individual is not updated in five consecutive generations, the algorithm is terminated early.

In the proposed $cEDA$, except for the parameters determined above, there are two important parameters (p_1 and p_2) to be determined: p_1 , which determines the number of better individuals selected from the current population for studying the parameters

of the probability model, and p_2 , which decides the number of the best individuals to reserve for the next generation. A preliminary numerical experiment is conducted to determine the two parameters. p_1 for testing is from the set $\{0.05, 0.10, 0.20, 0.40\}$, and p_2 is from the set $\{0.01, 0.05, 0.10, 0.20\}$.

In the preliminary experiments, the number of jobs is set to 20, the $cEDA$ is performed 5 times, and the amount of resources consumed is recorded each time. After completing the experiments, the average amount of resources consumed is calculated. The computational results are shown in Table 5. The combination $p_1 = 0.20$ and $p_2 = 0.20$ yields the optimal resource consumption.

5.3. Comparative approaches

5.3.1. Comparison of $cEDA_{c1}$ and $cEDA_{c2}$

In this section, two $cEDAs$ that adopt different copula functions (C_1 and C_2) are compared. The parameters of $cEDA_{c2}$ are the same as those of $cEDA_{c1}$ set forth above. In $cEDA_{c2}$, the unique parameters θ_a and θ_b are equal to 1 and 1.05, respectively. In each execution of $cEDA_{c1}$ and $cEDA_{c2}$, the solution and the running time obtained are recorded. In order to speed up all the algorithms involved in this paper under the configuration of 200 jobs, the following rule is implemented. When the best individual is not updated in three consecutive generations, the algorithm is terminated.

The meaning of columns in Tables 6 to 10 can be described as follows. Columns 1, 2 and 3 represent the size of the jobs, the workload of the jobs and the deadline, respectively. Columns 4 to 7 are the statistics of the amount of resources consumed in the ten runs of $cEDA_{c1}$, which are mean, SD, best and worst values. Columns 9 to 12 report the average, SD, best value and worst value of the amount of resource consumed among the 10 runs of $cEDA_{c2}$, respectively. Columns 8 and 13 report the running time (RT, s) of two approaches. Column 14 reports the degree of improvement (as shown in Eq. (26)) of $cEDA_{c1}$ compared to $cEDA_{c2}$ in terms of solution quality. And Column 15 in Table 10 represents the average reduction (as shown in Eq. (27)) obtained by using $cEDA_{c1}$ compared to $cEDA_{c2}$ in terms of RT.

The comparison results in Tables 6 to 9 illustrate that the $cEDA_{c1}$ is better than $cEDA_{c2}$ for the problem in small-scale and medium-scale instances. Compared to $cEDA_{c2}$ in terms of solution quality, the degree of improvement of $cEDA_{c1}$ is 74.90% for 10 jobs, 84.32% for 20 jobs, 83.25% for 50 jobs and 89.48% for 100 jobs. In large-scale instances (200 jobs), $cEDA_{c1}$ can obtain satisfactory solutions at the cost of 16% of the time, compared to $cEDA_{c2}$. The reason

Table 5

Results of the preliminary experiments.

p_1	p_2	Amount of resources consumed	p_1	p_2	Amount of resources consumed
0.40	0.01	382.53	0.40	0.05	354.71
0.40	0.10	324.69	0.40	0.20	291.30
0.40	0.40	290.73	0.20	0.01	297.27
0.20	0.05	297.44	0.20	0.10	285.90
0.20	0.20	282.77	0.10	0.01	303.25
0.10	0.05	296.52	0.10	0.10	303.86
0.05	0.01	362.73	0.05	0.05	365.95

Table 6Performance comparisons between $cEDA_{c1}$ and $cEDA_{c2}$ for 10 jobs.

Job size	Workload	Deadline	$cEDA_{c1}$					$cEDA_{c2}$					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1, 8]	20	42.87	5.92	36.54	57.80	0.38	124.76	86.41	55.09	318.59	0.59	86.54%
			43.03	4.91	37.53	53.51	0.42	192.83	150.61	60.18	425.81	0.63	89.89%
			43.58	2.97	40.40	50.87	0.41	126.63	87.88	61.04	354.27	0.71	87.70%
U[1,5]	U[1, 8]	15	77.1	6.31	70.08	90.37	0.36	189.37	111.56	102.36	421.65	0.57	81.71%
			78.71	6.72	73.12	92.28	0.34	214.23	120.57	111.33	442.04	0.45	82.19%
			75.84	4.31	70.95	82.68	0.32	162.95	76.04	118.02	372.15	0.60	79.62%
U[1,5]	U[1,16]	20	64.8	10.27	53.76	83.86	0.43	103.83	25.03	75.92	158.96	0.81	59.24%
			61.86	6.05	53.63	71.39	0.43	158.10	85.49	93.07	396.07	0.63	84.38%
			60.34	6.55	53.67	73.26	0.44	146.72	76.84	83.45	348.07	0.57	82.66%
U[1,5]	U[1,16]	15	105.83	2.71	102.33	110.19	0.39	276.02	82.05	192.99	463.45	0.36	77.16%
			110.6	6.92	103.71	124.35	0.38	255.11	88.85	175.32	443.97	0.49	75.09%
			108.24	5.82	103.18	121.73	0.4	271.53	64.26	178.95	430.02	0.40	74.83%
U[1,15]	U[1,8]	20	74.94	9.84	57.02	87.57	0.63	211.39	92.66	111.15	450.51	0.59	83.37%
			76.55	13.92	57.07	107.37	0.59	180.88	69.49	119.68	362.74	0.58	78.90%
			72.84	11.44	54.67	95.72	0.74	191.95	58.71	138.16	334.79	0.53	78.24%
U[1,15]	U[1,8]	15	964.43	252.14	658.25	1478.47	0.21	1212.81	361.43	775.54	1981.34	0.21	51.32%
			874.15	209.19	605.13	1285.95	0.27	1239.97	249.01	789.12	1630.72	0.23	46.39%
			735.09	81.45	640.23	922.63	0.36	1330.25	317.33	890.92	1794.06	0.23	59.03%
U[1,15]	U[1,16]	20	108.4	9.19	91.74	118.36	0.29	238.71	88.15	145.73	413.92	0.39	73.81%
			104.15	7.99	89.38	119.55	0.35	212.19	52.59	149.11	348.70	0.44	70.13%
			104.48	6.13	97.71	117.10	0.33	224.44	70.53	166.85	422.32	0.44	75.26%
U[1,15]	U[1,16]	15	890.87	112.48	720.79	1056.32	0.68	2794.48	296.98	2440.38	3378.63	0.23	73.63%
			939.72	269.75	689.84	1664.04	0.64	2654.36	539.98	1664.04	3456.41	0.25	72.81%
			936.32	336.76	719.17	1904.91	0.67	2737.49	367.59	2302.47	3575.21	0.22	73.81%
	Average		281.45	57.91	220.00	415.43	0.44	643.79	150.84	458.37	946.85	0.46	74.90%

Table 7Performance comparisons between $cEDA_{c1}$ and $cEDA_{c2}$ for 20 jobs.

Job size	Workload	Deadline	cEDA _{c1}					cEDA _{c2}					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	62.69	5.22	56.11	74.87	1.40	448.67	299.32	95.91	924.78	2.24	93.22%
			62.24	3.27	57.74	67.33	1.26	329.32	176.07	100.43	537.20	2.30	88.41%
			59.90	1.13	58.14	61.59	1.25	383.36	272.96	104.09	940.67	2.08	93.63%
U[1,5]	U[1,8]	15	52.77	3.73	46.11	58.89	2.14	356.94	300.9	124.39	949.39	2.54	94.44%
			55.1	7.09	48.74	72.17	2.01	486.91	310.01	113.71	934.21	1.92	94.10%
			57.01	6.39	49.01	69.71	2.02	432.92	244.06	131.35	777.08	1.90	92.66%
U[1,5]	U[1,16]	20	103.42	2.34	98.35	106.26	1.25	519.92	230.96	207.91	925.98	1.70	88.83%
			104.1	2.89	100.17	111.31	1.22	445.37	279.66	183.09	949.01	2.11	89.03%
			101.9	2.41	98.18	106.70	1.40	412.90	172.28	216.17	622.02	2.06	83.62%
U[1,5]	U[1,16]	15	109.04	11.30	97.62	135.16	1.81	472.92	208.78	223.89	936.92	1.70	88.36%
			116.67	14.26	96.79	140.54	1.62	413.6	151.86	215.19	630.06	1.98	81.48%
			107.24	8.47	100.87	126.57	1.83	518.35	299.15	230.42	1008.70	1.78	89.37%
U[1,15]	U[1,8]	20	121.4	9.48	111.11	141.72	2.01	575.00	236.42	267.29	941.68	1.50	87.11%
			120.7	14.47	106.20	159.64	2.15	487.49	231.81	210.45	945.96	2.27	87.24%
			117.3	9.15	109.39	135.62	2.27	533.55	241.10	283.03	963.45	1.62	87.83%
U[1,15]	U[1,8]	15	92.05	5.98	84.59	105.58	1.94	348.32	162.95	231.20	793.67	2.48	88.40%
			90.85	8.11	81.36	104.54	1.98	364.88	196.62	210.14	905.37	2.36	89.97%
			88.98	6.13	79.34	97.08	2.07	523.79	291.40	228.52	971.74	2.13	90.84%
U[1,15]	U[1,16]	20	326.22	26.36	295.21	393.03	1.26	908.54	99.51	734.56	1093.08	0.79	70.16%
			310.23	20.73	280.91	352.18	1.76	856.68	109.56	667.54	1024.03	0.73	69.70%
			304.83	15.5	276.04	324.14	1.83	902.67	106.06	718.26	1062.59	0.87	71.31%
U[1,15]	U[1,16]	15	338.68	14.11	300.20	352.93	1.47	882.63	149.51	646.71	1056.74	0.74	67.95%
			332.09	41.06	304.04	450.53	1.85	912.07	105.01	724.85	1052.42	0.86	68.45%
			330.34	13.46	302.26	353.29	1.65	875.6	106.21	679.49	1016.86	0.79	67.51%
	Average		148.57	10.54	134.94	170.89	1.73	558.02	207.59	314.52	915.15	1.73	84.32%

Table 8Performance comparisons between $cEDA_{c_1}$ and $cEDA_{c_2}$ for 50 jobs.

Job size	Workload	Deadline	$cEDA_{c_1}$					$cEDA_{c_2}$					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	93.72	4.22	89.15	105.73	22.39	1147.50	630.35	310.87	1862.00	19.60	91.83%
			81.88	3.54	76.37	87.97	23.05	1143.10	609.60	252.83	1957.50	18.32	92.84%
			122.70	8.32	115.98	146.49	28.36	719.36	537.47	311.37	1827.50	23.90	82.94%
U[1,5]	U[1,8]	15	186.67	4.15	175.42	190.12	20.04	1138.80	450.87	588.35	1887.60	13.70	83.61%
			314.04	12.76	299.11	338.79	28.68	1357.40	300.49	953.54	1867.70	10.88	76.86%
			215.43	6.05	203.82	222.86	25.74	1244.10	603.62	736.98	2661.50	13.88	82.68%
U[1,5]	U[1,16]	20	827.22	22.14	790.46	863.95	28.75	2755.10	156.63	2507.20	2989.30	4.98	69.98%
			353.98	20.56	327.73	402.54	34.18	1679.50	278.27	1193.90	2264.40	12.23	78.92%
			117.47	5.77	108.31	129.66	21.74	1130.70	670.58	325.89	2350.20	19.02	89.61%
U[1,5]	U[1,16]	15	111.54	4.86	102.57	117.26	17.66	1191.40	489.98	321.37	1875.20	16.10	90.64%
			152.03	4.06	144.70	156.60	14.81	907.11	515.66	427.04	1835.40	26.60	83.24%
			159.93	5.06	152.80	172.41	15.23	944.58	485.80	437.78	1693.70	23.36	83.07%
U[1,15]	U[1,8]	20	195.70	11.79	186.41	226.56	20.31	1092.20	555.86	536.27	1947.90	24.50	82.08%
			345.22	11.16	331.39	368.69	33.89	1605.60	374.64	1149.20	2461.80	10.93	78.50%
			509.31	19.93	484.15	550.14	25.85	2494.60	286.01	1843.80	2776.80	5.21	79.58%
U[1,15]	U[1,8]	15	335.30	6.73	325.22	347.61	22.00	1923.60	391.08	1271.20	2649.30	9.49	82.57%
			89.50	2.08	86.55	92.59	16.32	1212.60	647.61	260.95	2119.80	13.39	92.62%
			97.76	3.21	93.86	105.80	21.97	1321.50	584.83	300.89	2315.70	12.05	92.60%
U[1,15]	U[1,16]	20	126.02	3.23	120.64	133.46	19.24	1115.80	608.30	360.44	1843.80	15.64	88.71%
			176.09	3.55	171.22	182.62	21.34	996.26	458.25	594.02	1940.90	21.77	82.33%
			159.28	2.49	155.98	164.09	27.59	1718.80	765.19	618.47	2634.80	11.14	90.73%
U[1,15]	U[1,16]	15	322.66	27.86	297.44	384.34	25.27	1633.90	468.04	1250.60	2642.90	14.97	80.25%
			614.73	23.89	586.11	662.06	20.05	2335.90	390.16	1586.80	2708.30	5.39	73.68%
			790.68	39.87	736.03	883.46	25.83	2474.10	273.14	1946.40	2831.50	5.86	68.04%
	Average		270.79	10.72	256.73	293.16	23.35	1470.10	480.52	836.93	2247.70	14.71	83.25%

Table 9Performance comparisons between $cEDA_{c_1}$ and $cEDA_{c_2}$ for 100 jobs.

Job size	Workload	Deadline	$cEDA_{c_1}$					$cEDA_{c_2}$					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	154.74	2.99	151.70	159.81	161.80	3051.30	889.96	1731.50	4297.40	40.93	94.93%
			162.85	6.34	150.08	173.51	182.65	3046.40	1256.40	565.49	4701.10	66.65	94.65%
			210.86	4.31	203.14	215.35	145.73	3411.60	1174.30	1059.70	4619.10	55.13	93.82%
U[1,5]	U[1,8]	15	249.26	5.42	238.72	258.15	150.30	3098.30	1125.10	1148.00	4809.70	73.44	91.96%
			337.37	10.04	320.42	354.42	213.95	3280.80	1115.20	1771.10	5311.30	74.58	89.72%
			647.26	9.98	631.85	663.29	215.61	4859.00	666.83	3772.90	5610.00	31.41	86.68%
U[1,5]	U[1,16]	20	1028.50	17.33	1002.50	1061.90	263.81	4793.30	597.20	3676.30	5561.60	25.07	78.54%
			773.47	23.98	730.81	816.90	211.39	4510.00	781.86	3144.40	5573.20	25.09	82.85%
			179.67	5.11	171.33	187.38	111.54	3983.70	862.02	2272.20	5414.60	26.55	95.49%
U[1,5]	U[1,16]	15	202.52	4.23	196.76	209.56	105.79	2987.20	999.66	1298.40	4856.90	54.62	93.22%
			182.58	3.03	175.89	185.50	130.11	3141.80	1266.60	1591.00	5315.80	56.19	94.19%
			226.18	4.58	221.17	236.23	140.12	3334.30	959.79	1301.00	5018.70	41.85	93.22%
U[1,15]	U[1,8]	20	374.77	7.66	362.13	386.85	216.69	3060.10	1010.20	1649.50	4685.80	58.52	87.75%
			468.95	11.05	457.38	497.48	183.89	3327.70	797.19	1964.80	4473.60	44.72	85.91%
			570.44	16.90	543.42	604.11	175.10	3850.30	817.97	2210.70	4675.30	48.62	85.19%
U[1,15]	U[1,8]	15	766.07	25.28	738.52	824.23	247.42	4762.30	956.36	2447.40	5547.80	24.62	83.91%
			128.64	3.68	123.15	134.38	161.78	3397.30	1063.80	1778.10	5294.00	39.99	96.21%
			142.58	2.35	139.22	146.38	135.31	3292.60	1156.30	916.51	4484.30	50.47	95.67%
U[1,15]	U[1,16]	20	204.78	3.36	200.48	211.93	173.96	3742.20	1152.50	1774.10	5386.60	31.48	94.53%
			241.29	5.74	235.05	254.23	175.62	3254.70	1357.30	1203.90	5290.80	51.66	92.59%
			562.39	11.24	550.07	592.72	168.72	4035.20	716.14	2692.80	4750.90	31.56	86.06%
U[1,15]	U[1,16]	15	513.47	25.09	481.98	563.77	190.64	3192.80	806.04	2095.60	4688.70	69.77	83.92%
			844.58	24.42	806.99	890.94	243.66	4574.80	536.78	3511.70	5403.60	27.05	81.54%
			594.29	23.28	558.74	626.36	263.83	3966.00	627.84	2681.20	4711.40	38.15	85.02%
	Average		406.98	10.73	391.31	427.31	182.06	3664.70	945.56	2010.80	5020.10	45.34	89.48%

maybe that the conciseness of copula function in $cEDA_{c_1}$ make it run fast. And this ability make it more suitable for large-scale time-sensitive instances.

$$Improvement = \frac{Average(cEDA_{c_2}) - Average(cEDA_{c_1})}{Average(cEDA_{c_2})} \cdot 100\% \quad (26)$$

$$Reduction = \frac{RT(cEDA_{c_2}) - RT(cEDA_{c_1})}{RT(cEDA_{c_2})} \cdot 100\% \quad (27)$$

To further study the effectiveness of different copula functions in the cEDA, a computational experiment and comparison are conducted using $cEDA_{c_1}$ and $cEDA_{c_2}$. In this experiment, the size of the jobs follows a uniform distribution $U[1, 5]$; the workload of

the jobs follows a uniform distribution $U[1, 8]$; and the deadline is equal to 2 times the number of jobs. $cEDA_{c_1}$ and $cEDA_{c_2}$ are performed 30 times each for the same instances. The stopping criterion of two cEDAs is only the maximum generation. Fig. 4 shows the results of the comparative experiment in terms of solution quality. The average, best and worst values as well as the SD obtained by $cEDA_{c_1}$ are all better than those obtained by $cEDA_{c_2}$ in small-scale instances. However, in small-scale instances, the running time of the two cEDAs is not much different. Those results indicate that Clayton copula function C_1 in $cEDA_{c_1}$ fits the correlation better than Clayton copula function C_2 in $cEDA_{c_2}$ in small-scale instances.

Table 10Performance comparisons between cEDA_{c1} and cEDA_{c2} for 200 jobs.

Job size	Workload	Deadline	cEDA _{c1}					cEDA _{c2}					Impr.	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	9803.8	630.01	8399.9	10207	2.17	9494.0	1894.8	5765.6	11318	34.51	−3.26%	93.71%
			10035	197.57	9545.1	10242	2.09	9531.7	1492.8	5773.2	11229	37.41	−5.27%	94.41%
			9940.9	497.2	8466.6	10196	3.25	8605.1	1780.9	5117.1	10896	45.50	−15.52%	92.86%
U[1,5]	U[1,8]	15	9923.2	377.67	8877.4	10227	3.04	8370.1	2398.1	4252.6	11134	46.62	−18.56%	93.48%
			10138	354.66	9437.3	10391	10.77	10439.0	701.0	9174.2	11205	32.59	2.88%	66.95%
			10108	485.74	9069.8	10470	8.40	9755.6	1148.3	8077.3	11412	44.66	−3.61%	81.19%
U[1,5]	U[1,16]	20	10362	102.34	10120	10506	17.51	9339.6	1795.8	5700.9	11473	51.25	−10.95%	65.83%
			10130	494.34	8962	10566	22.34	10845.0	1242.7	7287.1	11617	32.21	6.59%	30.64%
			9736.2	545.26	8668.8	10211	3.49	9034.2	2175.4	5343.2	11205	41.15	−7.77%	91.52%
U[1,5]	U[1,16]	15	9953.8	389.43	8809.5	10219	3.11	9389.4	1654.9	5214.6	11021	40.67	−6.01%	92.35%
			9990.3	172.58	9576.1	10240	3.07	9576.5	1689.3	6647.6	11004	42.29	−4.32%	92.74%
			10079	71.87	9902.1	10160	3.04	9248.7	1198.3	7001.9	11021	38.63	−8.98%	92.13%
U[1,15]	U[1,8]	20	10140	332.62	9250.3	10398	6.34	9919.8	1178.3	7563.8	11135	52.16	−2.22%	87.85%
			10230	232.42	9584.8	10443	7.07	10081.0	1638.6	6824.6	11292	37.11	−1.48%	80.95%
			10347	214.55	9752.9	10514	17.13	10027.0	1462.4	6422.1	11478	36.97	−3.19%	53.67%
U[1,15]	U[1,8]	15	10117	448.95	9087.5	10456	14.49	9808.6	1471.3	6187.5	11418	40.10	−3.14%	63.87%
			10045	119.99	9811.4	10157	1.96	9261.8	2007.9	5436.7	11208	43.77	−8.45%	95.52%
			9884.9	401.73	8944.3	10157	2.63	9732.1	1373.1	6198.8	11109	55.29	−1.57%	95.24%
U[1,15]	U[1,16]	20	10021	207.57	9568.6	10212	3.47	9592.6	1397.7	7008.9	11006	50.47	−4.46%	93.12%
			9929.3	249.24	9259.4	10130	3.92	8886.9	1154.0	6504.4	10675	67.20	−11.73%	94.17%
			10129	394.57	9207.7	10453	5.77	9041.6	2318.8	5056.2	10783	76.60	−12.03%	92.47%
U[1,15]	U[1,16]	15	10184	389.4	9357.3	10493	12.20	8768.2	1923.0	5748.0	11562	89.39	−16.15%	86.35%
			10281	220.63	9672.1	10460	11.75	9852.6	1523.3	6324.6	11340	58.71	−4.35%	79.99%
			10311	223.71	9825.3	10547	19.24	10532.0	902.4	8482.4	11490	53.31	2.10%	63.91%
	Average		10076	323.09	9298.2	10336	7.84	9547.2	1563.5	6379.7	11210	47.86	−5.89%	83.61%

Table 11

Performance comparisons between the cEDA and GA for 10 jobs.

Job size	Workload	Deadline	cEDA					GA					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	38.19	2.70	34.99	44.63	0.43	397.55	23.86	356.78	431.93	0.46	91.16%
			43.30	10.61	34.36	65.40	0.43	370.39	60.4	250.53	428.81	0.50	89.90%
			39.38	2.35	35.97	43.13	0.38	397.26	46.27	333.51	459.05	0.54	91.42%
U[1,5]	U[1,8]	15	61.55	4.15	56.72	69.45	0.42	400.79	40.46	319.03	457.14	0.52	86.54%
			63.36	9.32	56.78	90.44	0.31	366.44	39.67	305.29	426.33	0.49	85.14%
			62.53	4.14	55.77	70.55	0.31	369.28	54.44	275.52	448.79	0.55	86.07%
U[1,5]	U[1,16]	20	62.74	4.95	56.48	72.42	0.32	383.07	56.25	282.98	504.85	0.51	87.57%
			61.60	3.61	58.46	70.57	0.32	350.43	44.81	271.85	420.64	0.49	85.36%
			65.87	12.47	57.56	92.71	0.33	372.26	54.90	229.66	435.74	0.47	84.88%
U[1,5]	U[1,16]	15	119.77	4.58	113.05	130.75	0.28	402.66	45.26	308.46	476.27	0.46	74.85%
			123.24	6.87	114.45	134.05	0.30	415.05	41.70	334.04	471.58	0.47	73.87%
			122.67	7.88	113.27	137.05	0.27	420.94	48.81	330.43	503.17	0.37	75.62%
U[1,15]	U[1,8]	20	89.02	7.20	75.79	99.46	0.47	398.94	36.88	350.08	448.69	0.49	80.16%
			90.15	7.99	71.05	100.33	0.46	387.86	40.96	292.47	438.60	0.52	79.44%
			89.64	7.71	79.52	103.07	0.41	409.98	46.13	317.38	466.66	0.43	80.79%
U[1,15]	U[1,8]	15	213.06	23.35	189.59	256.59	0.49	454.85	45.88	407.32	544.64	0.57	60.88%
			202.33	18.02	181.48	230.32	0.62	518.40	23.08	485.77	562.10	0.44	64.00%
			205.14	22.16	177.65	255.71	0.63	498.63	42.69	398.82	555.48	0.37	63.07%
U[1,15]	U[1,16]	20	78.48	7.83	64.35	96.49	0.53	380.13	38.04	321.64	439.86	0.60	82.16%
			86.47	10.24	74.16	108.52	0.54	404.57	35.46	355.61	469.24	0.53	81.57%
			84.16	16.48	70.03	123.83	0.50	395.21	43.10	342.09	465.89	0.52	81.94%
U[1,15]	U[1,16]	15	1639.57	538.24	1234.1	3201.2	0.56	3557.95	597.98	2555.14	4282.89	0.63	61.71%
			1500.32	116.26	1285.77	1662.26	0.62	3509.31	866.81	2192.67	4745.3	0.83	68.38%
			1731.21	499.57	1381.17	3204.91	0.53	3788.27	686.97	2936.61	5243.76	0.52	66.99%
	Average		286.41	56.20	236.36	435.99	0.44	806.26	127.54	606.40	1005.31	0.51	78.48%

5.3.2. Comparison of the cEDA and GA

In this section, a comparison between the GA proposed by Caraffa et al. [71] and cEDA is conducted. The parameters of the GA (except for the selection operator) are introduced in Caraffa et al. [71]. The proposed selection operator in our experiment is tournament selection. In addition, the number of individuals in the population and the stopping criterion are the same as for the cEDA.

Tables 11 to 15 show the experimental results obtained by the cEDA and GA for these instances in the case of 10, 20, 50, 100 and 200 jobs, respectively. Columns 4, 5, 6, 7 and 8 report the average, standard deviation (SD), best and worst value of the amount of resources consumed, and running time (RT) among the 10 runs of the cEDA, respectively. Columns 9, 10, 11, 12 and 13

report the average, SD, and best value and worst value of the amount of resource consumed, and RT among the 10 runs of the GA, respectively. Column 14 reports the average improvement (as shown in Eq. (28)) obtained by using the cEDA compared to the GA in terms of solution quality. In Table 15, the meaning of the first 14 columns is in accordance with Table 14. And Column 15 in Table 15 represents the average reduction (as shown in Eq. (29)) obtained by using cEDA compared to GA in terms of RT.

$$\text{Improvement} = \frac{\text{Average(GA)} - \text{Average(cEDA)}}{\text{Average(GA)}} \cdot 100\% \quad (28)$$

$$\text{Reduction} = \frac{\text{RT(GA)} - \text{RT(cEDA)}}{\text{RT(GA)}} \cdot 100\% \quad (29)$$

Table 12

Performance comparisons between the cEDA and GA for 20 jobs.

Job size	Workload	Deadline	cEDA					GA					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	49.23	5.27	44.04	64.03	2.03	882.65	57.69	761.46	948.94	1.71	94.81%
			52.58	7.90	45.50	68.43	2.07	865.36	47.45	807.89	930.34	1.63	94.35%
			51.29	5.29	44.86	59.88	2.18	847.03	92.01	683.6	961.76	1.74	94.67%
U[1,5]	U[1,8]	15	74.94	4.28	70.93	86.26	1.36	883.88	45.14	807.59	946.55	1.58	92.08%
			78.45	6.13	72.97	94.11	1.28	883.15	25.50	843.72	927.32	1.53	91.54%
			75.67	3.38	71.30	83.62	1.30	849.11	41.63	786.37	937.97	1.71	91.93%
U[1,5]	U[1,16]	20	92.84	6.13	86.88	106.44	1.46	862.16	52.96	783.14	960.40	1.63	90.33%
			90.15	6.50	80.42	100.87	1.55	858.54	39.42	763.66	906.47	1.55	90.05%
			86.00	4.97	80.29	94.20	1.73	893.62	41.06	828.33	951.15	1.63	90.96%
U[1,5]	U[1,16]	15	105.25	2.47	101.75	109.87	1.28	878.19	71.29	692.17	976.18	1.59	89.22%
			103.74	1.71	101.02	106.58	1.31	842.42	83.45	675.43	960.14	1.80	89.20%
			106.32	5.17	97.91	114.15	1.26	835.13	56.33	709.33	920.51	1.89	88.45%
U[1,15]	U[1,8]	20	199.41	14.62	181.44	229.28	2.04	875.49	59.62	807.32	977.58	2.15	79.60%
			202.76	13.77	184.92	232.13	1.75	902.53	51.66	814.85	969.69	1.94	79.09%
			195.22	19.07	173.25	235.11	2.53	914.66	42.72	831.97	981.47	1.86	80.11%
U[1,15]	U[1,8]	15	153.28	7.45	141.24	166.50	2.18	915.92	56.22	828.74	1019.86	1.87	84.97%
			143.69	12.53	124.73	166.12	2.81	906.11	89.93	702.82	1062.35	2.00	86.47%
			158.77	15.39	140.6	178.35	1.98	902.94	45.95	853.14	982.15	1.87	83.83%
U[1,15]	U[1,16]	20	150.41	12.96	134.94	173.46	1.95	893.79	60.46	774.00	1012.45	1.84	85.14%
			148.69	10.92	139.42	178.51	2.35	877.13	66.71	734.66	962.81	1.94	84.56%
			149.03	13.56	138.58	181.80	2.13	891.03	83.72	776.45	1051.61	1.68	85.83%
U[1,15]	U[1,16]	15	359.63	24.05	318.45	398.56	1.62	994.68	67.19	908.63	1092.42	1.74	67.08%
			355.31	19.46	306.59	376.72	1.86	975.55	78.28	808.85	1054.94	1.92	66.32%
			356.13	30.95	287.55	389.94	1.75	956.28	48.53	912.82	1034.03	2.00	65.56%
	Average		147.45	10.58	132.07	166.46	1.82	891.14	58.54	787.37	980.38	1.78	85.26%

Table 13

Performance comparisons between the cEDA and GA for 50 jobs.

Job size	Workload	Deadline	cEDA					GA					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	80.10	1.96	77.60	84.54	15.81	2396.30	51.72	2330.00	2488.60	18.93	96.66%
			107.61	3.08	103.79	114.60	16.43	2381.60	64.56	2271.00	2454.30	18.43	95.48%
			113.44	3.37	109.13	119.94	22.69	2378.10	61.91	2276.60	2475.50	19.16	95.23%
U[1,5]	U[1,8]	15	142.36	7.65	132.58	158.83	22.04	2375.10	86.22	2240.60	2535.70	18.74	94.01%
			263.47	5.00	257.97	274.02	25.90	2431.30	46.20	2354.90	2530.70	19.51	89.16%
			318.18	12.64	292.73	339.02	38.92	2382.60	130.89	2161.60	2569.60	21.49	86.65%
U[1,5]	U[1,16]	20	736.61	19.80	704.69	779.20	25.95	2726.60	86.73	2585.20	2821.10	19.93	72.98%
			1124.00	49.28	1058.90	1247.10	19.88	3083.20	147.61	2762.70	3283.00	21.54	63.54%
			81.62	3.01	78.26	88.54	21.47	2396.70	59.39	2304.30	2482.10	18.98	96.60%
U[1,5]	U[1,16]	15	93.98	6.54	89.31	108.53	25.04	2384.40	57.57	2281.30	2456.00	18.46	96.06%
			141.23	3.79	136.33	147.26	20.57	2461.70	50.41	2352.00	2520.80	17.97	94.26%
			135.51	3.51	129.47	140.55	21.49	2405.90	76.14	2229.10	2527.50	19.05	94.37%
U[1,15]	U[1,8]	20	235.38	5.56	227.19	247.86	20.47	2433.20	83.10	2280.30	2595.00	19.61	90.33%
			245.17	8.97	232.39	265.67	22.52	2390.80	33.49	2346.20	2456.70	20.04	89.75%
			455.27	15.55	433.16	485.59	34.17	2527.40	143.03	2372.50	2762.10	21.09	81.99%
U[1,15]	U[1,8]	15	383.25	13.23	363.06	404.08	27.62	2393.90	56.52	2293.00	2491.90	21.52	83.99%
			72.22	0.92	70.56	74.19	16.57	2402.30	56.99	2309.60	2525.90	18.63	96.99%
			94.48	1.59	91.77	96.78	16.15	2382.40	63.77	2289.20	2478.50	17.69	96.03%
U[1,15]	U[1,16]	20	147.46	3.38	142.03	152.96	16.84	2379.50	70.29	2221.50	2476.50	18.38	93.80%
			140.78	2.97	137.45	148.73	19.87	2435.70	55.47	2373.70	2530.40	18.42	94.22%
			243.18	7.62	231.79	255.87	23.53	2446.70	74.74	2346.90	2568.00	19.21	90.06%
U[1,15]	U[1,16]	15	187.31	4.78	181.13	196.65	29.80	2456.20	76.26	2336.30	2598.60	17.17	92.37%
			287.83	9.42	274.60	305.26	28.66	2424.50	130.07	2272.70	2745.70	18.92	88.13%
			348.90	19.98	320.81	385.68	29.16	2383.90	104.60	2177.30	2515.90	18.02	85.36%
	Average		257.47	8.90	244.86	275.90	23.40	2452.50	77.82	2323.70	2578.70	19.20	89.92%

As shown in Tables 11 to 15, the solutions obtained by the cEDA are generally better than those obtained by the GA in terms of solutions quality. In addition, compared to GA, cEDA has prominent advantages on small-scale and medium-scale instances. And on large-scale instances, cEDA can obtain satisfactory solutions in much less time, compared to GA. The reason is that there are very time consuming crossover and mutation operators in GA.

5.3.3. Comparison of cEDA and PSO

The proposed cEDA is compared to PSO in this section. The parameters of the PSO algorithm are adopted from Muthuswamy

et al. [9]. PSO algorithm is a swarm intelligent search algorithm, in which the individuals are referred to as particles. The velocity-updating approach for the particles can be described as follows:

$$v_{jv}(t+1) = w(t) \cdot v_{jv}(t) + c_1 r_1 \cdot (LBest_{jv} - u_{jv}(t)) + c_2 r_2 \cdot (GBest - u_{jv}(t)), \quad (30)$$

where r_1 and r_2 follows a uniform distribution $U(0, 1)$ and, $w(t)$, c_1 and c_2 are parameters in the PSO approach. And $w(t)$ is updated using $w(t+1) = \alpha \cdot w(t)$, where α is the decrement factor. In this comparison of experiments, parameter w is initially set to 0.9 and, c_1 and c_2 are equal to 1 [9].

Table 14

Performance comparisons between the cEDA and GA for 100 jobs.

Job size	Workload	Deadline	cEDA					GA					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	152.49	3.67	147.83	157.56	139.62	5008.90	90.86	4902.00	5151.40	126.04	96.96%
			163.78	4.04	158.04	170.94	161.16	5083.60	69.55	4939.80	5217.30	130.93	96.78%
			197.94	2.52	194.76	203.46	129.31	5069.90	71.32	4911.10	5188.60	131.71	96.10%
U[1,5]	U[1,8]	15	213.77	3.31	209.02	219.23	171.65	5032.50	118.84	4810.00	5191.60	131.32	95.75%
			425.73	18.35	386.50	445.24	230.28	5222.50	90.84	4986.70	5335.80	143.68	91.85%
			435.26	9.14	416.11	450.26	309.88	5114.20	121.54	4866.80	5274.80	139.69	91.49%
U[1,5]	U[1,16]	20	784.99	15.54	764.04	810.73	205.59	5370.70	188.51	5091.70	5639.10	138.62	85.38%
			593.50	21.68	570.26	631.38	199.21	5065.40	90.52	4934.10	5241.30	153.81	88.28%
			143.00	4.32	139.90	152.98	146.38	5055.00	97.70	4926.10	5245.90	126.18	97.17%
U[1,5]	U[1,16]	15	136.89	2.80	132.12	141.26	167.89	5079.70	112.73	4932.00	5289.30	134.19	97.31%
			292.77	5.50	284.99	304.35	114.35	5039.90	87.46	4934.60	5237.70	131.40	94.19%
			242.12	6.81	233.86	258.84	173.25	5018.30	64.07	4898.60	5114.10	139.96	95.18%
U[1,15]	U[1,8]	20	553.50	14.84	536.60	584.29	273.73	5116.80	112.78	4925.50	5336.50	160.09	89.18%
			467.83	16.14	452.24	506.95	194.11	5083.10	122.24	4908.70	5288.90	160.82	90.80%
			938.78	39.70	893.54	1026.10	262.58	5329.30	250.08	4919.90	5621.60	140.00	82.38%
U[1,15]	U[1,8]	15	466.78	10.43	446.78	482.19	350.55	5169.60	107.17	5035.60	5345.20	155.37	90.97%
			130.08	2.18	126.16	133.44	150.92	5030.10	72.97	4936.00	5147.80	119.57	97.41%
			121.89	2.41	118.61	128.07	153.54	5020.30	69.42	4871.10	5108.80	124.99	97.57%
U[1,15]	U[1,16]	20	260.80	4.87	252.63	268.54	164.72	4992.30	137.46	4742.40	5207.20	137.21	94.78%
			211.53	3.23	206.45	217.11	128.71	5030.30	112.83	4745.20	5190.90	141.04	95.80%
			648.06	12.79	631.48	670.91	219.95	5183.70	249.44	4834.90	5618.30	149.09	87.50%
U[1,15]	U[1,16]	15	385.47	13.44	366.48	415.32	341.20	5159.10	118.50	4922.90	5314.30	149.14	92.53%
			570.49	10.22	553.87	588.34	280.44	5037.80	145.68	4766.60	5254.20	160.37	88.68%
			682.69	18.84	648.76	717.69	211.56	5186.80	241.45	4809.90	5591.10	155.52	86.84%
			Average	384.17	10.28	369.63	403.55	203.36	5104.10	122.66	4898.00	5298.00	140.86

Table 15

Performance comparisons between the cEDA and GA for 200 jobs.

Job size	Workload	Deadline	cEDA					GA					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	10052	310.94	9154.1	10284	2.61	10760	396.94	10238	11339	393.04	6.59%	99.34%
			10099	105.41	9905.9	10253	1.65	10617	303.96	10263	11289	384.05	4.88%	99.57%
			9815.8	974.40	6900.5	10268	4.69	10823	334.54	10354	11337	414.92	9.31%	98.87%
U[1,5]	U[1,8]	15	9903.9	290.69	9341.2	10171	3.78	10655	416.88	10250	11300	408.34	7.05%	99.07%
			10311	126.13	10052	10525	6.32	10962	404.75	10480	11548	449.51	5.93%	98.59%
			10208	243.76	9518.9	10421	7.78	10734	294.11	10363	11453	436.05	4.90%	98.22%
U[1,5]	U[1,16]	20	10502	181.76	9975.7	10631	22.03	11493	84.14	11360	11617	556.11	8.63%	96.04%
			10290	297.57	9421.6	10522	14.07	11135	392.43	10380	11488	490.11	7.59%	97.13%
			9984.7	170.06	9554.6	10140	2.80	10723	437.53	10185	11391	389.98	6.89%	99.28%
U[1,5]	U[1,16]	15	9926.1	295.67	9054.5	10092	2.57	10540	213.03	10196	10936	387.52	5.83%	99.34%
			9989.7	275.61	9183.6	10144	3.48	10826	356.39	10382	11316	409.14	7.73%	99.15%
			10026	187.02	9533	10246	3.65	10525	269.61	10370	11315	371.09	4.74%	99.02%
U[1,15]	U[1,8]	20	10235	276.37	9435.6	10407	8.46	10854	336.67	10485	11581	418.09	5.70%	97.98%
			10384	88.72	10230	10522	11.64	11073	417.34	10480	11574	445.48	6.22%	97.39%
			10297	123.55	9976.7	10430	9.04	11050	368.59	10597	11521	420.26	6.81%	97.85%
U[1,15]	U[1,8]	15	10389	125.13	10174	10583	18.66	11519	75.36	11399	11636	499.50	9.81%	96.26%
			10005	246.10	9279.6	10186	2.33	10877	370.82	10341	11343	353.60	8.02%	99.34%
			9886.2	516.02	8423.1	10249	2.85	10687	373.46	10318	11335	352.84	7.49%	99.19%
U[1,15]	U[1,16]	20	9989.5	202.92	9404.1	10145	2.61	10697	327.29	10324	11320	353.85	6.62%	99.26%
			9980.9	321.37	9072	10243	3.39	10763	368.11	10296	11283	367.45	7.27%	99.08%
			10067	459.82	8890.5	10383	6.51	10927	460.13	10341	11503	399.14	7.87%	98.37%
U[1,15]	U[1,16]	15	10293	156.94	9889.5	10462	8.63	10981	430.76	10379	11523	424.13	6.27%	97.97%
			10265	193.91	9828.2	10507	13.83	11341	334.94	10404	11635	455.65	9.49%	96.96%
			10196	286.68	9361.7	10386	10.64	11082	341.41	10443	11463	425.81	8.00%	97.50%
	Average		10129	269.02	9398.4	10342	7.25	10902	337.88	10443	11419	416.90	7.07%	98.26%

Tables 16 to 20 provide the experimental results obtained by the cEDA and PSO for these instances in the case of 10, 20, 50, 100 and 200 jobs, respectively. Column 14 reports the average improvement (as shown in Eq. (31)) obtained using the cEDA compared to the PSO in terms of solution quality. And Column 15 in Table 20 represents the average reduction (as shown in Eq. (32)) obtained by using cEDA compared to PSO in terms of RT.

$$\text{Improvement} = \frac{\text{Average(PSO)} - \text{Average(cEDA)}}{\text{Average(PSO)}} \cdot 100\% \quad (31)$$

$$\text{Reduction} = \frac{\text{RT(PSO)} - \text{RT(cEDA)}}{\text{RT(PSO)}} \cdot 100\% \quad (32)$$

Tables 16 to 19 show that the solutions obtained by the cEDA are better than those obtained by PSO in small-scale and medium-scale instances. However, the running time of cEDA is more than PSO in small-scale and medium-scale instances. In large-scale instances, cEDA can obtain the solutions which are only 1.28% less than the quality of solutions obtained by PSO in only half the time.

5.3.4. Comparison of cEDA and MIMIC_c

The proposed cEDA is compared to the MIMIC_c algorithm in this section. The MIMIC_c algorithm proposed by Larraanaga and Lozano [57] can address the continuous optimization problem. The parameters of the MIMIC_c algorithm are identical to the proposed cEDA.

Table 16

Performance comparisons between the cEDA and PSO for 10 jobs.

Job size	Workload	Deadline	cEDA					PSO					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	43.51	5.16	37.71	53.7	0.49	405.85	50.5	309.24	493.18	0.05	89.28%
			71.19	4.09	67.61	79.68	0.41	408.57	30.67	358.58	462.6	0.06	82.58%
			70.27	4.69	64.14	77.3	0.43	413.06	36.59	324.9	468.26	0.05	82.99%
U[1,5]	U[1,8]	15	120.03	6.29	111.39	129.44	0.37	431.66	32.09	371.59	508.74	0.07	72.19%
			68.67	9.23	59.28	89.27	0.58	402.19	26.18	362.42	462.04	0.05	82.93%
			206.56	7.3	196.07	222.7	0.62	443.82	34.48	379.62	496.53	0.09	53.46%
U[1,5]	U[1,16]	20	119.47	8.94	109.19	135.86	0.72	432.88	27.38	384.85	483.56	0.07	72.40%
			1390.7	227.63	1169.4	2010.6	0.77	4131.6	500.85	3476.5	4830.3	0.19	66.34%
			37.55	4.25	32.36	45.98	0.53	405.94	44.72	312.81	476.59	0.05	90.75%
U[1,5]	U[1,16]	15	71.4	2.8	66.84	74.44	0.42	412.59	38	338.1	483.43	0.06	82.70%
			53.59	3.57	49.74	59.17	0.53	412.97	44.12	319.39	476.2	0.06	87.02%
			126.32	3.67	121.4	132.51	0.39	418.94	30.91	360.54	452.81	0.07	69.85%
U[1,15]	U[1,8]	20	69.28	9.9	58.21	91.4	0.73	420.97	23.01	382.95	474.06	0.06	83.54%
			210.72	14.73	189.12	248.13	0.68	475.67	35.27	408.06	546.23	0.09	55.70%
			94.7	10.8	80.42	119.18	0.85	431.42	20.65	391.43	456.33	0.08	78.05%
U[1,15]	U[1,8]	15	8473.6	881.19	7409.8	10645	0.91	18519	2171.5	15489	21631	0.23	54.24%
			35.21	2.29	32.49	39.59	0.49	397.92	55.57	299.49	480.18	0.05	91.15%
			95.82	8.94	86.68	121.01	0.5	398.68	29.13	354.22	446.16	0.07	75.97%
U[1,15]	U[1,16]	20	60.82	4.02	54.83	68.06	0.43	409.27	29.37	358.11	457.49	0.06	85.14%
			120.93	3.64	114.42	126.34	0.32	419.81	16.25	392.04	453.33	0.06	71.19%
			64.12	2.6	60.15	69.17	0.68	442.8	30.7	401.65	494.24	0.06	85.52%
U[1,15]	U[1,16]	15	269.72	17.5	246.69	308.97	0.59	510.77	41.07	454.58	582.54	0.09	47.19%
			129.87	8.55	116.43	146.65	0.65	428.7	40.52	332.2	484.11	0.07	69.71%
			244.43	15.85	216.73	279.44	0.79	516.43	30.21	485.52	578.16	0.14	52.67%
	Average		510.35	52.82	447.96	640.57	0.58	1337.2	142.49	1127	1549.1	0.08	74.27%

Table 17

Performance comparisons between the cEDA and PSO for 20 jobs.

Job size	Workload	Deadline	cEDA					PSO					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	68.75	5.62	64.95	82.90	1.71	886.06	34.79	818.87	938.12	0.14	92.24%
			75.95	2.70	72.05	80.94	2.54	900.97	54.90	810.94	976.17	0.15	91.57%
			85.96	6.46	80.54	104.26	2.49	869.38	44.77	797.72	941.24	0.16	90.11%
U[1,5]	U[1,8]	15	112.51	3.49	106.29	116.81	1.93	931.32	27.85	865.93	977.28	0.16	87.92%
			118.53	9.61	108.16	139.68	2.74	902.10	54.23	802.03	960.62	0.16	86.86%
			169.64	10.02	157.39	186.53	2.70	917.08	35.78	840.84	968.02	0.23	81.50%
U[1,5]	U[1,16]	20	199.29	20.04	163.57	237.72	3.44	898.44	51.94	810.53	987.81	0.23	77.82%
			253.65	21.29	219.76	300.00	3.55	913.04	74.59	800.47	1035.10	0.23	72.22%
			46.38	3.51	40.78	53.11	2.71	890.76	29.62	837.85	937.56	0.14	94.79%
U[1,5]	U[1,16]	15	76.00	1.47	72.81	78.40	1.85	878.32	62.36	807.04	1001.30	0.16	91.35%
			97.42	5.06	89.19	106.07	2.04	874.89	59.73	780.38	994.69	0.13	88.87%
			100.78	5.75	92.89	109.91	2.80	882.65	38.30	791.89	923.77	0.15	88.58%
U[1,15]	U[1,8]	20	156.46	5.79	146.39	163.51	2.13	884.40	59.71	797.35	996.12	0.20	82.31%
			270.31	14.06	253.70	295.53	2.87	917.16	70.80	779.09	1041.50	0.23	70.53%
			240.69	9.96	225.38	260.26	2.27	897.95	56.37	805.80	1009.90	0.24	73.20%
U[1,15]	U[1,8]	15	354.88	15.25	330.49	380.51	2.37	971.33	52.04	878.29	1059.30	0.27	63.47%
			53.15	6.01	47.53	66.69	2.69	876.75	85.84	740.60	982.89	0.13	93.94%
			83.27	4.79	75.55	88.71	2.12	872.91	53.82	775.34	961.62	0.13	90.46%
U[1,15]	U[1,16]	20	125.33	5.10	116.60	135.63	1.72	836.46	97.84	661.53	968.30	0.15	85.02%
			102.52	2.27	98.90	105.95	1.53	870.57	117.59	563.38	1026.20	0.14	88.22%
			176.30	15.15	152.52	204.71	2.72	902.55	62.21	755.99	978.02	0.19	80.47%
U[1,15]	U[1,16]	15	115.47	8.78	102.66	137.89	3.89	892.21	83.89	679.95	990.04	0.18	87.06%
			211.58	12.12	191.75	231.77	2.72	850.47	86.51	723.37	964.67	0.22	75.12%
			1010.80	314.61	608.20	1414.50	1.67	1369.30	102.72	1212.20	1582.30	0.63	26.19%
	Average		179.40	21.21	150.75	211.75	2.47	911.96	62.43	797.39	1008.40	0.20	81.66%

Tables 21 to 25 provide the experimental results obtained by the cEDA and *MIMIC_c* for these instances in the case of 10, 20, 50, 100 and 200 jobs, respectively. Column 14 reports the average improvement (as shown in Eq. (33)) obtained using the cEDA compared to the *MIMIC_c* in terms of solution quality. And Column 15 in Table 25 represents the average reduction (as shown in Eq. (34)) obtained by using cEDA compared to *MIMIC_c* in terms of RT.

$$\text{Improvement} = \frac{\text{Average}(\text{MIMIC}_c) - \text{Average}(\text{cEDA})}{\text{Average}(\text{MIMIC}_c)} \cdot 100\% \quad (33)$$

$$\text{Reduction} = \frac{\text{RT}(\text{MIMIC}_c) - \text{RT}(\text{cEDA})}{\text{RT}(\text{MIMIC}_c)} \cdot 100\% \quad (34)$$

As can be observed from Table 21, the solutions obtained by the cEDA are better than those obtained by *MIMIC_c* in all of 24 instances for 10 jobs instance. Specifically, the degree of improvement of cEDA, compared to *MIMIC_c* in terms of solution quality, is an average of 33.36% in much less running time. In addition, the SD values of the solutions produced by the cEDA are smaller in 20 of the 24 configurations when compared to *MIMIC_c*. Table 22 presents that the solutions obtained by the cEDA are better than those obtained by the *MIMIC_c* in 23 of the 24 configurations for 20 jobs instance. Compared to *MIMIC_c*, cEDA achieves a 15.89% average improvement rate in terms of solution quality, and cEDA takes only about half the running time of *MIMIC_c*. In addition, the SD values of the solutions produced by the cEDA are smaller in

Table 18

Performance comparisons between the cEDA and PSO for 50 jobs.

Job size	Workload	Deadline	cEDA					PSO					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	86.17	2.83	81.21	90.47	21.08	2329.00	102.67	2071.00	2458.50	0.56	96.30%
			84.08	1.85	80.75	86.97	19.32	2331.30	94.60	2201.00	2481.40	0.58	96.39%
			126.64	2.67	121.67	130.16	14.77	2286.70	114.45	2039.00	2451.20	0.65	94.46%
U[1,5]	U[1,8]	15	142.39	4.18	136.31	149.32	18.91	2290.80	102.46	2152.40	2530.20	0.66	93.78%
			244.30	7.26	232.09	253.51	21.32	2392.60	105.39	2150.50	2529.00	1.01	89.79%
			331.99	11.52	311.94	358.11	20.13	2385.60	68.55	2289.80	2478.30	1.27	86.08%
U[1,5]	U[1,16]	20	519.56	17.10	488.29	550.55	17.31	2378.30	90.89	2145.90	2468.50	1.26	78.15%
			545.14	23.82	501.28	583.09	25.35	2349.90	78.67	2215.00	2486.00	1.69	76.80%
			94.00	3.97	89.05	102.07	12.97	2343.30	94.22	2234.60	2531.70	0.58	95.99%
U[1,5]	U[1,16]	15	117.27	2.92	112.64	121.40	14.24	2373.90	87.20	2186.70	2513.40	0.56	95.06%
			119.94	5.17	114.31	131.96	17.67	2420.10	114.81	2164.90	2597.40	0.65	95.04%
			154.80	4.82	146.06	160.81	18.32	2337.40	97.25	2195.10	2531.60	0.71	93.38%
U[1,15]	U[1,8]	20	248.10	7.59	241.91	265.93	24.25	2334.60	113.55	2129.20	2546.80	1.03	89.37%
			366.77	13.92	346.19	386.66	36.88	2332.20	67.80	2250.90	2442.10	1.39	84.27%
			457.87	16.92	419.78	479.45	36.72	2384.80	80.07	2243.40	2535.70	1.38	80.80%
U[1,15]	U[1,8]	15	324.61	7.05	315.45	340.35	27.47	2342.10	53.54	2259.80	2437.90	1.14	86.14%
			93.87	2.67	87.43	96.53	16.80	2355.50	93.60	2202.40	2517.50	0.52	96.02%
			93.21	2.53	89.40	97.17	18.67	2384.40	73.58	2287.10	2557.70	0.49	96.09%
U[1,15]	U[1,16]	20	127.57	4.64	119.75	137.70	21.17	2297.10	96.65	2079.80	2410.50	0.67	94.45%
			118.42	3.60	113.25	125.06	28.32	2328.00	74.25	2196.20	2420.10	0.66	94.91%
			327.52	14.03	303.57	351.24	18.78	2390.70	122.31	2200.80	2543.00	1.14	86.30%
U[1,15]	U[1,16]	15	383.84	15.81	356.84	403.47	21.97	2316.70	117.76	2145.40	2457.10	1.43	83.43%
			307.90	11.32	282.39	327.91	27.71	2361.30	95.02	2243.50	2501.20	1.17	86.96%
			341.55	9.12	326.63	359.56	33.77	2349.70	129.22	2019.50	2519.90	1.32	85.46%
	Average		239.90	8.22	225.76	253.73	22.25	2349.80	94.52	2179.30	2497.80	0.94	89.81%

Table 19

Performance comparisons between the cEDA and PSO for 100 jobs.

Job size	Workload	Deadline	cEDA					PSO					Improvement
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
U[1,5]	U[1,8]	20	152.30	2.20	148.98	155.43	125.19	4848.80	116.50	4639.60	5049.60	2.36	96.86%
			168.98	2.92	165.55	174.48	133.23	4707.90	152.66	4414.40	4931.30	2.60	96.41%
			227.24	4.75	220.59	235.61	166.80	4857.90	221.17	4518.90	5138.50	3.29	95.32%
U[1,5]	U[1,8]	15	212.76	7.88	207.32	234.72	182.46	4690.20	324.55	4136.70	5122.20	2.99	95.46%
			553.70	14.53	534.65	588.27	185.69	4776.30	171.26	4456.70	5071.70	6.31	88.41%
			509.14	7.76	497.35	523.10	223.59	4755.30	351.31	4029.20	5194.30	6.24	89.29%
U[1,5]	U[1,16]	20	540.75	22.95	502.28	574.90	218.88	4840.40	113.19	4698.70	5070.00	5.27	88.83%
			926.35	44.86	843.00	987.83	214.63	4877.30	165.44	4514.80	5130.90	9.90	81.01%
			135.88	2.64	132.13	140.43	115.17	4786.10	182.08	4503.10	5025.50	2.46	97.16%
U[1,5]	U[1,16]	15	160.38	3.75	154.94	167.02	148.88	4872.30	145.62	4568.90	5082.60	2.38	96.71%
			201.61	4.85	190.08	208.83	121.07	4864.40	167.66	4596.20	5160.70	2.83	95.86%
			253.36	7.87	244.16	271.11	172.91	4971.00	109.55	4820.90	5202.00	3.25	94.90%
U[1,15]	U[1,8]	20	696.81	16.52	664.99	717.27	199.35	4939.30	130.92	4758.00	5137.90	7.57	85.89%
			432.92	15.91	414.72	465.40	194.16	4845.80	271.72	4128.50	5086.50	5.38	91.07%
			581.25	17.36	547.68	609.29	271.30	4736.40	241.68	4274.00	5038.90	6.37	87.73%
U[1,15]	U[1,8]	15	743.78	23.47	712.97	782.22	194.75	4834.70	169.72	4377.40	5029.90	7.44	84.62%
			139.73	4.54	133.96	147.56	110.18	4877.40	143.11	4639.70	5132.80	2.01	97.14%
			133.82	2.74	129.78	139.08	119.42	4750.10	260.06	4027.70	5029.20	2.00	97.18%
U[1,15]	U[1,16]	20	227.25	3.35	223.73	235.19	114.52	4805.40	136.36	4526.30	4961.20	2.48	95.27%
			217.42	5.68	207.70	227.01	123.14	4819.80	139.96	4621.80	4990.80	2.48	95.49%
			526.63	12.86	509.27	549.31	173.83	4896.90	100.96	4721.50	5065.90	5.58	89.25%
U[1,15]	U[1,16]	15	519.33	7.29	506.07	532.83	221.91	4773.10	244.09	4436.80	5194.90	5.50	89.12%
			540.99	13.73	521.24	560.74	242.24	4735.00	187.66	4438.20	5052.20	5.96	88.58%
			689.68	23.09	663.43	746.07	191.20	4728.40	191.63	4367.40	4960.90	7.03	85.41%
	Average		395.50	11.40	378.19	415.57	173.52	4816.30	184.95	4467.30	5077.50	4.57	91.79%

19 of the 24 configurations when compared to *MIMIC_c*. Table 23 shows that the solutions obtained by the cEDA are better than those obtained by the *MIMIC_c* in 17 of the 24 configurations for 50 jobs instance. Compared to *MIMIC_c*, cEDA achieves a 1.31% average improvement rate in terms of solution quality, and the average running time of cEDA is reduced by 57.75%. As can be observed from Table 24, there is no significant difference between cEDA and *MIMIC_c* in terms of solutions' quality. On the quality of solutions, *MIMIC_c* is only 0.8% better than cEDA. However, compared with *MIMIC_c*, the proposed cEDA approach reduces the running time by 47.95%.

From Tables 21 to 25, it can be concluded that the proposed cEDA approach outperforms *MIMIC_c* in small-scale instances in terms of quality of solutions and running time. And in medium-scale and large-scale instances, the proposed cEDA approach can obtain the solutions, which are very close to the solutions produced by *MIMIC_c*, in much less time. In other words, the proposed cEDA approach can reduce greatly the running time, at the expense of a little bit of the quality of solutions.

Fig. 5 illustrates that with the number of jobs increasing, the reduction of RT generally becomes larger. Meanwhile, the quality of solutions cEDA obtained is still very close to *MIMIC_c*.

Table 20

Performance comparisons between the cEDA and PSO for 200 jobs.

Job size	Workload	Deadline	cEDA					PSO					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	10125	72.35	10008	10266	2.66	10233	184.07	9974.9	10493	8.76	1.05%	69.63%
			10023	229.34	9351.9	10169	2.70	10197	243.62	9742.5	10522	8.64	1.71%	68.75%
			9817.6	445.13	8848.1	10158	3.64	10092	195.53	9804.1	10526	10.05	2.72%	63.78%
U[1,5]	U[1,8]	15	10133	82.58	9949	10211	3.71	9936.7	252.09	9446	10311	11.57	−1.98%	67.93%
			10247	320.77	9580.9	10529	9.10	9983.5	547.98	8737.2	10739	19.46	−2.64%	53.24%
			10335	93.32	10202	10500	8.47	10110	222.63	9787.2	10411	22.88	−2.23%	62.98%
U[1,5]	U[1,16]	20	10482	366.72	9625.9	10764	24.48	10009	275.35	9611.9	10431	38.95	−4.72%	37.15%
			10283	142.91	10011	10507	13.58	9649	334.39	8963.3	10373	26.06	−6.57%	47.89%
			9895.7	320.71	9153.5	10255	2.11	10256	165.18	9942.3	10509	7.60	3.51%	72.24%
U[1,5]	U[1,16]	15	10016	180.28	9674.7	10227	2.55	10255	139.41	9999.4	10511	8.43	2.33%	69.75%
			9939.5	189.37	9593.5	10187	3.21	10294	198.60	10051	10723	8.63	3.44%	62.80%
			10067	70.66	9939.5	10192	2.70	10151	260.88	9781	10502	9.37	0.83%	71.18%
U[1,15]	U[1,8]	20	10457	65.35	10364	10558	15.57	9705	350.91	9096.6	10141	31.12	−7.75%	49.97%
			10119	557.18	8486.9	10465	10.76	9681.1	586.58	8967.1	10604	22.22	−4.53%	51.58%
			10278	210.20	9748.2	10454	14.67	9689.6	206.94	9429.4	10143	27.43	−6.07%	46.52%
U[1,15]	U[1,8]	15	10309	98.58	10172	10461	13.71	9840.6	364.86	9074.7	10463	22.42	−4.76%	38.85%
			9905.4	463.44	8991.2	10289	2.15	10303	143.26	10123	10651	6.91	3.86%	68.89%
			9935.3	233.64	9365.7	10158	2.54	10214	206.38	9706.7	10434	7.09	2.73%	64.17%
U[1,15]	U[1,16]	20	9923.8	282.11	9096.2	10141	3.28	10098	160.85	9856.4	10364	9.96	1.73%	67.07%
			9864.9	487.83	8528.6	10203	2.88	10206	208.05	9915.1	10563	8.51	3.34%	66.16%
			10289	194.41	9746.9	10502	8.53	9925.3	505.97	9050.2	10819	18.64	−3.66%	54.24%
U[1,15]	U[1,16]	15	10300	127.83	10043	10522	7.77	9997.6	400.13	9075.6	10457	19.36	−3.03%	59.87%
			10378	92.69	10225	10494	15.93	9744.6	405.71	9165.3	10371	33.10	−6.50%	51.87%
			10295	115.20	10143	10494	14.65	9936.4	240.40	9360.7	10230	23.03	−3.61%	36.39%
	Average		10142	226.77	9618.8	10363	7.97	10021	283.32	9527.5	10470	17.09	−1.28%	53.35%

Table 21Performance comparisons between the cEDA and *MIMIC_c* for 10 jobs.

Job size	Workload	Deadline	cEDA					MIMIC _c					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	40.12	6.94	33.57	53.32	0.50	84.12	18.69	59.45	123.96	0.95	52.31%	47.37%
			60.43	3.42	56.73	67.91	0.41	103.30	22.32	73.86	145.14	0.92	41.50%	55.43%
			61.68	3.46	59.39	71.43	0.45	115.53	15.08	83.52	129.19	0.84	46.61%	46.43%
U[1,5]	U[1,8]	15	143.94	5.31	136.92	152.92	0.37	165.51	12.49	147.94	193.40	0.77	13.03%	51.95%
			65.71	15.06	39.89	88.20	0.77	112.38	19.45	93.17	163.49	0.99	41.53%	22.22%
			617.14	232.07	458.72	1099	1.05	784.92	67.95	674.76	907.35	1.26	21.38%	16.67%
U[1,5]	U[1,16]	20	164.67	13.33	145.67	184.16	0.92	245.26	24.30	211.26	276.23	1.05	32.86%	12.38%
			344.01	23.66	319.22	407.03	0.93	477.44	26.01	436.78	519.37	1.24	27.95%	25.00%
			42.24	2.84	39.17	47.63	0.46	94.19	12.91	76.97	117.77	0.90	55.15%	48.89%
U[1,5]	U[1,16]	15	69.72	3.50	65.15	76.12	0.45	111.48	17.52	91.15	156.78	0.87	37.46%	48.28%
			71.98	4.42	66.34	79.91	0.42	109.85	11.74	94.24	132.40	0.95	34.47%	55.79%
			90.74	3.85	86.12	97.03	0.43	124.15	14.97	96.82	143.61	0.91	26.91%	52.75%
U[1,15]	U[1,8]	20	62.51	6.07	52.41	71.49	0.65	117.26	12.10	94.45	134.08	0.92	46.69%	29.35%
			5413.30	453.03	4650.30	6459.40	0.96	7098.60	410.13	6430.20	7660.30	1.19	23.74%	19.33%
			719.08	141.52	576.88	1073.00	0.99	879.20	59.98	789.87	974.44	1.18	18.21%	16.10%
U[1,15]	U[1,8]	15	335.38	47.38	287.09	449.86	1.00	490.59	58.04	429.28	628.10	1.19	31.64%	15.97%
			39.62	3.19	36.25	44.66	0.47	92.17	25.87	51.85	146.01	0.91	57.01%	48.35%
			67.43	2.48	63.77	71.66	0.48	105.49	18.81	78.55	141.36	0.98	36.08%	51.02%
U[1,15]	U[1,16]	20	61.02	2.66	56.59	64.74	0.48	107.83	9.67	90.12	127.97	0.94	43.41%	48.94%
			124.49	4.07	119.08	129.89	0.38	146.38	10.87	129.87	164.86	0.82	14.95%	53.66%
			86.08	7.76	74.07	98.50	0.65	126.10	16.41	90.90	147.33	0.99	31.74%	34.34%
U[1,15]	U[1,16]	15	182.22	13.01	160.82	205.81	0.69	227.59	11.36	206.43	242.21	0.97	19.94%	28.87%
			260.56	23.02	226.04	306.41	1.06	357.88	28.34	324.15	396.27	1.12	27.19%	5.36%
			1283.30	75.36	1199.90	1443.10	0.77	1580.80	163.37	1345.30	1967.30	1.26	18.82%	38.89%
	Average		433.64	45.73	375.42	535.13	0.66	577.42	45.35	508.37	655.78	1.01	33.36%	34.65%

The benchmarks of well-known shop scheduling problems in Taillard [72] are all combinatorial optimization problems. For demonstrating the effectiveness of cEDA in well-known problems, the cEDA is tested in three benchmark optimization functions, such as Sphere function (f_1), Ackley function (f_2), etc. And the dimensions of all three functions are set to 10, 20 and 30 in this experiment. The definition of the three functions are given in Eq. (35). The experimental results are in Table 26. And the results show that the cEDA outperforms *MIMIC_c*. The running time of cEDA can be greatly reduced, compared to *MIMIC_c*. This result is consistent with the experimental result in this paper. The reduction in the running

time of cEDA can be attributed to its simple structure.

$$\begin{aligned}
 f_1(\mathbf{x}) &= f_1(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^2, \quad x_i \in [-100, 100] \\
 f_2(\mathbf{x}) &= f_2(x_1, \dots, x_n) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} \\
 &\quad + 20 + e, \quad x_i \in [-5, 5] \\
 f_3(\mathbf{x}) &= f_3(x_1, x_2, \dots, x_n) = \sum_{i=1}^n [(x_1 - x_i^2)^2 + (x_i - 1)^2], \\
 &\quad x_i \in [-10, 10]
 \end{aligned} \tag{35}$$

Table 22Performance comparisons between the cEDA and *MIMIC_c* for 20 jobs.

Job size	Workload	Deadline	cEDA					MIMIC _c					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	50.18	3.63	46.11	58.27	2.93	75.23	11.47	53.60	90.25	5.78	33.30%	49.31%
			77.50	4.52	68.76	86.37	1.98	87.25	5.37	78.28	98.78	5.16	11.18%	61.63%
			93.38	3.97	87.61	101.93	1.75	109.05	11.62	97.44	132.70	5.43	14.37%	67.77%
U[1,5]	U[1,8]	15	102.61	5.42	91.99	110.29	2.02	125.58	9.45	115.82	148.82	5.13	18.29%	60.62%
			79.24	5.12	72.85	89.91	2.77	110.93	11.80	90.14	128.64	5.96	28.57%	53.52%
			257.17	17.53	232.67	293.35	3.25	298.64	30.53	258.05	343.77	5.94	13.89%	45.29%
U[1,5]	U[1,16]	20	544.00	43.63	475.28	606.83	3.31	560.04	32.73	503.13	608.39	5.76	2.86%	42.53%
			2519.20	1490.50	1597	6061	4.43	2139.80	135.83	1994.80	2492.10	7.91	−17.73%	43.99%
			49.55	2.07	45.54	52.40	3.31	73.72	8.58	53.90	83.79	5.93	32.79%	44.18%
U[1,5]	U[1,16]	15	63.09	5.69	57.79	78.09	2.57	89.97	13.72	72.53	115.64	5.81	29.88%	55.77%
			81.94	2.01	78.58	86.38	1.60	96.32	6.32	86.72	110.25	4.51	14.93%	64.52%
			117.73	2.44	112.20	120.83	1.88	130.60	7.39	119.52	141.94	5.08	9.85%	62.99%
U[1,15]	U[1,8]	20	134.93	6.34	122.59	148.76	2.75	163.89	8.47	154.35	184.71	5.01	17.67%	45.11%
			166.75	8.04	151.29	183.37	3.59	200.93	17.00	182.41	235.63	5.62	17.01%	36.12%
			417.25	22.37	388.73	466.62	2.43	440.38	18.56	406.42	475.18	5.06	5.25%	51.98%
U[1,15]	U[1,8]	15	225.50	9.40	210.93	238.76	3.03	268.60	9.40	256.74	285.46	6.31	16.05%	51.98%
			54.66	1.13	53.23	56.95	1.61	73.49	5.70	67.45	82.26	5.29	25.62%	69.57%
			64.15	4.46	58.54	74.70	3.02	86.40	5.54	72.63	92.54	6.23	25.75%	51.52%
U[1,15]	U[1,16]	20	90.88	3.13	87.91	98.95	2.46	116.83	9.59	106.38	137.54	5.07	22.21%	51.48%
			93.09	9.36	83.61	115.26	2.69	118.56	12.44	104.13	145.67	6.62	21.48%	59.37%
			256.23	9.17	246.70	272.17	2.43	274.35	12.54	256.14	297.76	5.26	6.60%	53.80%
U[1,15]	U[1,16]	15	225.60	27.16	194.88	284.17	4.18	246.40	17.50	215.71	272.98	6.54	8.44%	36.09%
			239.43	13.54	221.33	262.27	3.33	276.50	17.46	246.56	301.36	6.32	13.41%	47.31%
			372.87	20.07	339.02	400.32	2.48	412.40	16.15	388.94	436.80	5.61	9.59%	55.79%
	Average		265.71	71.70	213.55	431.17	2.74	274.00	18.13	249.24	310.13	5.72	15.89%	52.10%

Table 23Performance comparisons between the cEDA and *MIMIC_c* for 50 jobs.

Job size	Workload	Deadline	cEDA					MIMIC _c					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	96.72	3.25	91.31	104.37	15.5	98.21	2.65	95.4	105.05	52.38	1.52%	70.41%
			83.17	3.4	79.61	90.93	23.9	86.99	5.78	79.97	96.85	58.02	4.39%	58.81%
			118.89	2.37	116.5	124.4	21.77	123.46	2.31	120.48	127.76	54.37	3.70%	59.96%
U[1,5]	U[1,8]	15	137.11	5.81	131.18	152.55	16.53	137.88	2.62	133.59	142.82	48.44	0.56%	65.88%
			438.06	20.41	413.29	482.51	20.22	430.16	12.8	404.03	446.74	49.17	−1.84%	58.88%
			307.89	7.85	298.81	323.47	17.78	306.66	9.32	295.33	326.99	46.88	−0.40%	62.07%
U[1,5]	U[1,16]	20	598.25	16.83	576.09	629.98	16.92	577.53	10.59	564.77	601.43	44.42	−3.59%	61.91%
			494.07	24.42	435.34	526.17	20.23	492.05	14.69	470.99	524.5	49.28	−0.41%	58.95%
			83.19	1.52	80.42	85.73	16.33	85.73	3.53	82.58	91.98	52.55	2.96%	68.92%
U[1,5]	U[1,16]	15	82.75	3.42	77.77	87.26	25.42	86.82	4.38	79.56	92.36	58.65	4.69%	56.66%
			139.99	2.99	135.76	144.52	20.73	143.47	3.87	138.61	149.91	53.73	2.43%	61.42%
			162.19	4.96	154.86	174.04	21.69	167.75	12.05	157.83	196.55	58.18	3.31%	62.72%
U[1,15]	U[1,8]	20	237.19	9.53	216.49	245.7	28.77	236.23	6	221.47	241.71	60.86	−0.41%	52.73%
			212.41	6.3	203.55	225.04	27.17	213.25	5.54	205.01	224.88	56.67	0.39%	52.06%
			314.04	8.89	300.51	325.5	29.9	306.71	12.42	285.75	329.44	64.34	−2.39%	53.53%
U[1,15]	U[1,8]	15	394.17	15.44	375.4	433.23	22.17	390.74	10.8	376.1	408.94	52.49	−0.88%	57.76%
			99.02	3.08	94.7	104.89	17.94	101.73	3.97	97.58	109.01	52.47	2.66%	65.81%
			100.99	3.01	95.4	105.93	18.52	105.1	4.27	99.3	114.28	52.9	3.91%	64.99%
U[1,15]	U[1,16]	20	113.31	5.49	108.68	127.34	24.56	118.13	3.45	113.84	123.71	57.86	4.08%	57.55%
			165.97	4.56	160.06	175.57	17.25	173.91	8.07	165.01	190.63	51.39	4.57%	66.43%
			255.69	13.16	239.17	277.69	36.09	257.26	8.72	239.05	269.61	62.11	0.61%	41.89%
U[1,15]	U[1,16]	15	366.16	11.53	350.41	380.84	22.36	366.22	6.32	353.62	377.9	51.5	0.02%	56.58%
			376.45	11.97	362.24	402.72	39.03	379.6	15.96	357.38	404.28	61.72	0.83%	36.76%
			727.77	17.65	704.61	752.48	32.52	732.96	12.83	716.8	758.65	58.99	0.71%	44.87%
	Average		254.39	8.66	241.76	270.12	23.05	254.94	7.62	243.92	269.00	54.56	1.31%	57.75%

5.4. Analysis of convergence trends

In this section, the convergence profiles of cEDA, GA, PSO and *MIMIC_c* on five different scale instances are given. For all instances, the job size comes from the uniform distribution of intervals [1,5], the workload comes from the uniform distribution of intervals [1,8], and the deadline is equal to 20. As can be seen from Fig. 6, compared with cEDA and *MIMIC_c*, GA and PSO converge prematurely especially for small-sized instances. Although both cEDA and *MIMIC_c* can maintain the diversity of the population well, the proposed cEDA approach can greatly reduce the running time as we stated before in Section 5.3.4.

6. Conclusion and future work

This paper examines a scheduling problem with a convex RCF. The aim of this paper is to determine an approximate optimal resource allocation vector \mathcal{R} of all jobs with the constraint $C_{max} \leq D$, to minimize the total amount of resources consumed. A mathematical model is implemented for the problem. To address the problem, a cEDA approach is used, and a new, advanced correction algorithm is proposed to amend the individuals that do not satisfy the deadline constraint.

The cEDA approach is rarely used in flow-shop scheduling problems in the literature even, though it can improve the learning efficiency of the EDA and reduce running time. A computational experiment is performed to study the effectiveness of different

Table 24Performance comparisons between the cEDA and *MIMIC_c* for 100 jobs.

Job size	Workload	Deadline	cEDA					MIMIC _c					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	160.17	3.75	153.73	166.68	126.20	160.56	3.25	155.40	166.29	332.66	0.24%	62.06%
			131.77	2.09	128.84	135.09	127.39	131.96	2.35	127.87	135.96	345.57	0.14%	63.14%
			149.01	3.20	143.69	154.85	121.06	150.20	2.74	143.96	155.78	322.13	0.79%	62.42%
U[1,5]	U[1,8]	15	170.52	2.34	167.30	174.08	112.13	172.22	3.93	166.85	180.64	292.21	0.99%	61.63%
			151.62	3.57	147.42	159.38	125.58	151.96	2.63	148.88	157.09	324.32	0.22%	61.28%
			172.78	3.80	167.18	180.45	107.93	170.21	2.94	166.35	176.96	337.63	−1.51%	68.03%
U[1,5]	U[1,16]	20	226.53	3.79	221.24	232.69	139.03	226.27	1.48	223.68	228.61	319.07	−0.11%	56.43%
			206.93	6.03	197.25	220.30	140.21	203.79	6.83	197.09	218.69	355.63	−1.54%	60.57%
			265.57	4.26	259.28	271.40	148.50	264.23	4.58	257.79	272.79	333.46	−0.51%	55.47%
U[1,5]	U[1,16]	15	273.40	9.27	258.90	285.96	114.07	274.63	4.26	267.87	282.69	275.78	0.45%	58.64%
			229.70	6.65	220.77	245.11	144.45	233.10	3.33	227.78	239.05	330.06	1.46%	56.24%
			210.81	3.49	204.92	216.83	135.84	213.07	3.15	207.89	217.21	304.97	1.06%	55.46%
U[1,15]	U[1,8]	20	388.79	12.58	365.63	407.86	210.96	382.98	11.30	365.63	407.73	364.68	−1.52%	42.15%
			445.05	2.99	441.95	452.05	240.01	444.27	8.20	431.53	459.52	413.33	−0.18%	41.93%
			393.23	5.69	385.82	402.07	212.16	381.91	7.87	366.95	392.52	427.65	−2.96%	50.39%
U[1,15]	U[1,8]	15	350.92	13.13	330.64	374.05	187.53	340.91	11.99	329.53	371.17	389.42	−2.94%	51.84%
			478.72	14.10	455.69	507.36	188.26	465.93	7.62	448.45	478.09	374.73	−2.75%	49.76%
			440.45	5.27	432.30	448.42	376.23	443.72	20.82	426.14	496.98	525.29	0.74%	28.38%
U[1,15]	U[1,16]	20	741.11	18.36	724.48	782.89	183.60	723.28	19.39	692.68	754.18	325.91	−2.47%	43.67%
			684.12	15.65	666.06	715.63	281.34	668.27	8.10	654.46	685.41	455.45	−2.37%	38.23%
			930.80	20.40	908.59	969.71	325.34	926.41	25.76	894.44	987.84	450.05	−0.47%	27.71%
U[1,15]	U[1,16]	15	724.85	18.74	683.78	746.79	261.52	691.97	17.10	669.69	724.46	484.98	−4.75%	46.08%
			548.03	11.49	522.98	570.94	349.52	543.75	19.14	520.99	575.17	491.06	−0.79%	28.82%
			977.89	23.32	938.73	1013.20	364.05	969.29	31.20	929.17	1013.10	497.25	−0.89%	26.79%
			Average	393.87	8.92	380.30	409.74	196.79	388.95	9.58	375.88	407.41	378.05	−0.82%

Table 25Performance comparisons between the cEDA and *MIMIC_c* for 200 jobs.

Job size	Workload	Deadline	cEDA					<i>MIMIC_c</i>					Improvement	RT Reduction
			Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT		
U[1,5]	U[1,8]	20	9980	134.57	9771.3	10164	2.50	9820.4	340.28	8944.8	10154	24.99	−1.63%	90.00%
			9921.7	232.77	9390.8	10121	2.94	9947.5	253.84	9191.7	10083	23.93	0.26%	87.71%
			10029	152.01	9617.4	10193	3.28	9789.5	386.44	8797.5	10116	26.13	−2.45%	87.45%
U[1,5]	U[1,8]	15	10034	102.44	9815.4	10195	3.33	9913.5	241.92	9229.2	10117	24.68	−1.21%	86.51%
			10294	115.89	10057	10435	6.18	10207	83.14	10062	10357	26.31	−0.85%	76.51%
			10352	86.19	10201	10489	11.94	10284	83.19	10093	10402	32.64	−0.66%	63.42%
U[1,5]	U[1,16]	20	10345	132.97	10137	10529	14.58	10244	179.94	9771	10408	38.47	−0.98%	62.10%
			10466	50.2	10369	10566	18.04	10285	278.83	9466.3	10434	43.39	−1.77%	58.42%
			10052	75.46	9931.3	10174	2.42	9884.7	313.15	9148.6	10101	24.04	−1.70%	89.93%
U[1,5]	U[1,16]	15	9977.2	112.23	9775.9	10143	2.38	9709.6	491.38	8481.6	10140	26.99	−2.76%	91.18%
			10066	97.17	9843.2	10171	3.52	10054	79.97	9901.3	10150	22.91	−0.12%	84.64%
			9892.3	374.82	8972.9	10187	3.53	9952.5	116.94	9704.4	10078	22.94	0.61%	84.61%
U[1,15]	U[1,8]	20	10203	393.73	9323	10487	11.22	10289	73.63	10146	10395	31.47	0.84%	64.35%
			10314	72.23	10206	10394	9.49	10135	257.55	9403.5	10326	32.42	−1.77%	70.73%
			10342	88.95	10198	10504	13.65	10172	341.31	9179.2	10378	38.99	−1.67%	64.99%
U[1,15]	U[1,8]	15	10403	95.64	10211	10576	16.55	10316	53.84	10218	10416	39.66	−0.84%	58.27%
			9952.1	339.71	9024.1	10284	2.56	9875.5	333.33	8890.8	10051	24.00	−0.78%	89.33%
			9808	617.69	8111.5	10202	2.97	10047	58.84	9958	10150	21.55	2.38%	86.22%
U[1,15]	U[1,16]	20	9897.5	447.81	8799.7	10231	3.55	10035	92.25	9811.2	10138	21.88	1.37%	83.78%
			10116	94.81	9922.3	10231	2.96	9932	277.96	9110.3	10087	23.47	−1.85%	87.39%
			10329	106.41	10026	10415	9.29	10154	269.94	9490.6	10355	32.44	−1.72%	71.36%
Average			10360	174.32	9878.4	10511	17.25	10166	316.43	9345.5	10416	43.46	−1.91%	60.31%
			10391	59.35	10291	10466	16.56	10291	274.49	9485.1	10496	42.26	−0.97%	60.81%
			10159	177.19	9751.1	10338	7.78	10064	239.79	9425.3	10254	29.93	−0.95%	74.02%

Table 26Performance comparison between the cEDA and *MIMIC_c* for well-known benchmark optimization functions.

Function	Dimension	cEDA					<i>MIMIC_c</i>					RT Reduction
		Average	SD	Best	Worst	RT	Average	SD	Best	Worst	RT	
f_1	10	0.01	0.02	0	0.06	0.04	22.73	42.85	0	126.56	0.18	77.78%
	20	0	0	0	0	0.48	1.7	0.51	0	1.7	1.59	69.81%
	30	0	0	0	0	3.17	0	0	0	0	13.56	76.62%
f_2	10	1.72	0	1.72	1.72	0.01	1.72	0	1.72	1.72	0.03	66.67%
	20	1.72	0	1.72	1.72	0.01	1.72	0	1.72	1.72	0.12	91.67%
	30	1.72	0	1.72	1.72	0.01	1.72	0	1.72	1.72	0.24	95.83%
f_3	10	0.63	0.99	0.01	3.56	0.02	2.23	2.53	0.07	8.97	0.14	85.71%
	20	1.22	2.98	0	10.12	0.05	1.20	1.41	0.03	4.86	0.84	94.05%
	30	1.78	2.16	0.1	7.68	0.08	3.01	5.11	0.01	16.6	2.36	96.61%

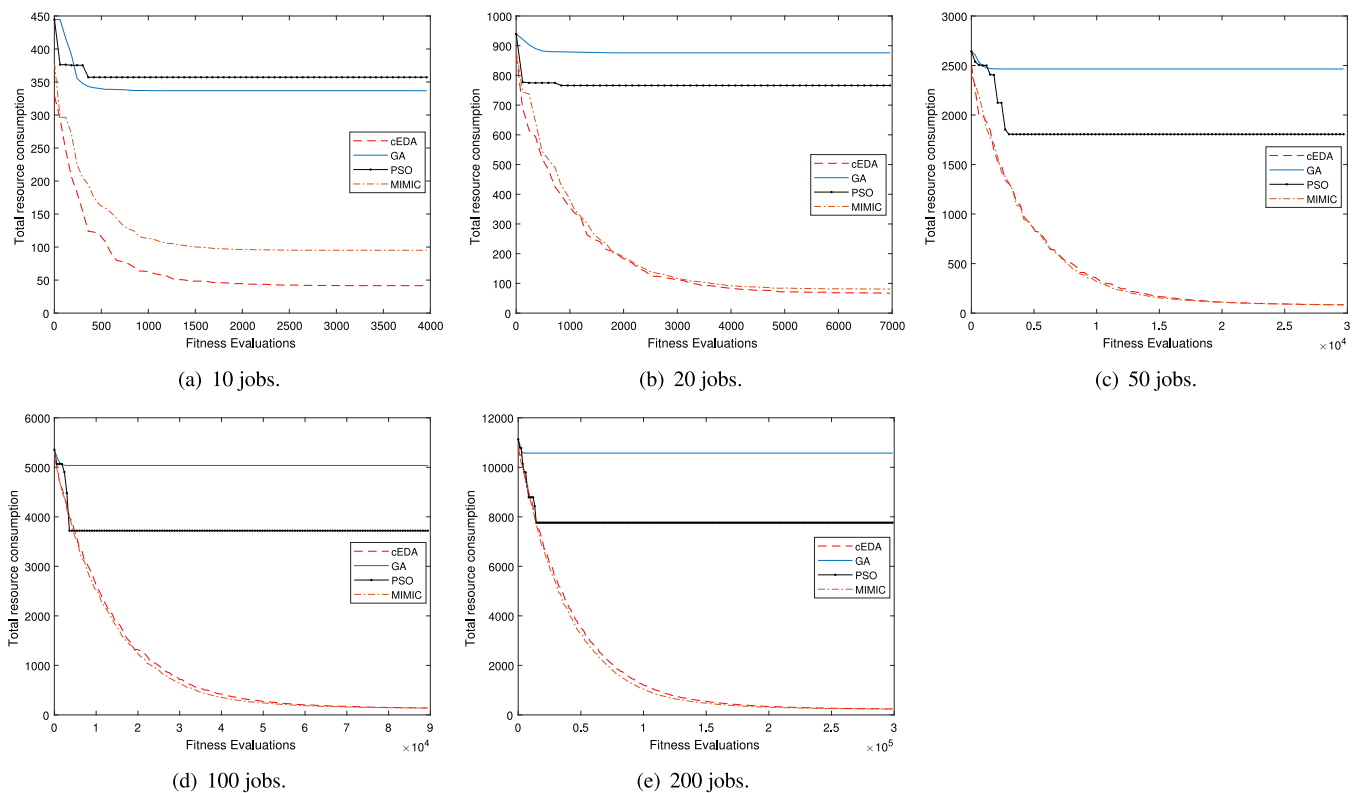


Fig. 6. Convergence trends of cEDA, GA, PSO and $MIMIC_c$ on the five different scale instances. Note that $MIMIC_c$ is abbreviated as MIMIC.

copula functions in the cEDA approach. The results illustrate that copula function C_1 is more effective than copula function C_2 in small-scale and medium-scale instances in terms of the quality of solutions and running time. To evaluate the performance of the cEDA approach in this problem, the GA, PSO and $MIMIC_c$ approaches are adopted for comparison. Computational results demonstrate that the cEDA has a competitive advantage over the GA, PSO and $MIMIC_c$ in small-scale instances in terms of the quality of solutions. In medium-scale instances, cEDA still outperforms GA and PSO, but the solution obtained by $MIMIC_c$ is very close to cEDA. The competitive advantage of cEDA decreases in large-scale instances. And $MIMIC_c$ and PSO can even obtain a little better solutions in some large-scale instances. However, the running time of cEDA can be greatly reduced in large-scale instances, compared to GA, PSO and $MIMIC_c$. The cEDA can give researchers more choices to balance the quality of solution and the running time in flow-shop scheduling problems.

The literature has rarely addressed flow-shop scheduling problems in which the resource consumption is the objective and the deadline is the constraint. However, the meaning of this type of problem is of great significance. The research of this problem can help to reduce resource costs (such as fossil fuel costs, financial outlays and manpower costs) of production and improve the efficiency of resource consumption. Considering the fact that few studies treat resource consumption as the objective in the flow shop scheduling problems, this research may fill this gap in the literature.

The cEDA approach proposed for the problem in this work can be extended to other flow-shop scheduling problems considering resource consumption. One possible direction for future study is to adapt cEDA to flow-shop scheduling problems with more than two machines, and the constraint of dynamic job arrival. Both problem settings are derived from reality and the complexity of the problem is greatly increased. Furthermore, it is necessary to find a copula function suitable for the problem characteristics, so as to achieve an expected performance.

Acknowledgment

The authors are very grateful to the Editor-in-Chief and the anonymous reviewers for their constructive comments and suggestions that have further helped to improve the quality and presentation of this paper. This research was supported by the National Natural Science Foundation of China grant number 71671168 and the Fundamental Research Funds for the Central Universities of China grant number 2013B18020204.

References

- [1] J. Müller, W. Mayer, A.Y. Oral, Z.B. Bahşi, M. Özer, M. Sezer, M.E. Aköz, Improving the resource efficiency of the german steel industry using material flow analysis, in: AIP Conference Proceedings, vol. 1653, AIP Publishing, 2015, p. 020074.
- [2] K. Dahlström, P. Ekins, Eco-efficiency trends in the UK steel and aluminum industries, *J. Ind. Ecol.* 9 (4) (2005) 171–188.
- [3] S. Henningsson, K. Hyde, A. Smith, M. Campbell, The value of resource efficiency in the food industry: a waste minimisation project in East Anglia, UK, *J. Cleaner Prod.* 12 (5) (2004) 505–512.
- [4] M. Pagotto, A. Halog, Towards a circular economy in Australian agri-food industry: An application of input-output oriented approaches for analyzing resource efficiency and competitiveness potential, *J. Ind. Ecol.* 20 (5) (2016) 1176–1186.
- [5] C.I.P. Martínez, Energy use and energy efficiency development in the German and Colombian textile industries, *Energy Sustain. Dev.* 14 (2) (2010) 94–103.
- [6] V. García, E. Pongrácz, P.S. Phillips, R.L. Keiski, From waste treatment to resource efficiency in the chemical industry: recovery of organic solvents from waters containing electrolytes by pervaporation, *J. Cleaner Prod.* 39 (2013) 146–153.
- [7] E. Alkaya, G.N. Demirel, Improving resource efficiency in surface coating/painting industry: practical experiences from a small-sized enterprise, *Clean Technol. Environ. Policy* 16 (8) (2014) 1565–1575.
- [8] Z.h. Jia, Y.-l. Zhang, J.Y.-T. Leung, K. Li, Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines, *Appl. Soft Comput.* 55 (2017) 226–237.
- [9] S. Muthuswamy, M.C. Velez-Gallego, J. Maya, M. Rojas-Santiago, Minimizing makespan in a two-machine no-wait flow shop with batch processing machines, *Int. J. Adv. Manuf. Technol.* 63 (1–4) (2012) 281–290.

- [10] J.H. Ahmadi, R.H. Ahmadi, S. Dasu, C.S. Tang, Batching and scheduling jobs on batch and discrete processors, *Oper. Res.* 40 (4) (1992) 750–763.
- [11] T.E. Cheng, G. Wang, Batching and scheduling to minimize the makespan in the two-machine flowshop, *IIE Trans.* 30 (5) (1998) 447–453.
- [12] A. Oulamara, Makespan minimization in a no-wait flow shop problem with two batching machines, *Comput. Oper. Res.* 34 (4) (2007) 1033–1050.
- [13] C.M. Martinez, X. Hu, D. Cao, E. Velenis, B. Gao, M. Wellers, Energy management in plug-in hybrid electric vehicles: Recent progress and a connected vehicles perspective, *IEEE Trans. Veh. Technol.* 66 (6) (2017) 4534–4549.
- [14] L. Zhang, X. Hu, Z. Wang, F. Sun, D.G. Dorrell, A review of supercapacitor modeling, estimation, and applications: A control/management perspective, *Renew. Sustain. Energy Rev.* 81 (2018) 1868–1878.
- [15] C.A. Hernandez-Aramburo, T.C. Green, N. Mugniot, Fuel consumption minimization of a microgrid, *IEEE Trans. Ind. Appl.* 41 (3) (2005) 673–681.
- [16] H. Holtkamp, G. Auer, S. Bazzi, H. Haas, Minimizing base station power consumption, *IEEE J. Sel. Areas Commun.* 32 (2) (2014) 297–306.
- [17] C.Y. Lee, R. Uzsoy, L.A. Martin-vega, Efficient algorithms for scheduling semiconductor burn-in operations, *Oper. Res.* 40 (4) (1992) 764–775.
- [18] L. Tang, H. Gong, J. Liu, F. Li, Bicriteria scheduling on a single batching machine with job transportation and deterioration considerations, *Naval Res. Logist.* 61 (4) (2014) 269–285.
- [19] S. Zhou, H. Chen, X. Li, Distance matrix based heuristics to minimize makespan of parallel batch processing machines with arbitrary job sizes and release times, *Appl. Soft Comput.* 52 (2017) 630–641.
- [20] M. Mathirajan, A.I. Sivakumar, A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor, *Int. J. Adv. Manuf. Technol.* 29 (9–10) (2006) 990–1001.
- [21] C.T.D. Ng, T.C.E. Cheng, M.Y. Kovalyov, Single machine batch scheduling with jointly compressible setup and processing times, *European J. Oper. Res.* 153 (1) (2004) 211–219.
- [22] D. Oron, Scheduling a batching machine with convex resource consumption functions, *Inform. Process. Lett.* 111 (19) (2011) 962–967.
- [23] F. Jolai, H. Kor, S.M. Hatefi, H. Iranmanesh, A genetic algorithm for makespan minimization in a no-wait flow shop problem with two batching machines, in: 2009 International Conference on Computer Engineering and Technology, Vol. II, Proceedings, 2009, pp. 238–242.
- [24] S.C. Zhou, X.P. Li, H.P. Chen, C. Guo, Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm, *Int. J. Prod. Res.* 54 (16) (2016) 4919–4937.
- [25] D. Shabtay, M. Kaspi, G. Steiner, The no-wait two-machine flow shop scheduling problem with convex resource-dependent processing times, *IIE Trans.* 39 (5) (2007) 539–557.
- [26] A. Janiak, W. Janiak, M. Lichtenstein, Resource management in machine scheduling problems: a survey, *Decis. Mak. Manuf. Serv. Vol. 1, No. 1–2* (2007) 59–89.
- [27] R.G. Vickson, Choosing the job sequence and processing times to minimize total processing plus flow cost on a single-machine, *Oper. Res.* 28 (5) (1980) 1155–1167.
- [28] R.G. Vickson, Two single machine sequencing problems involving controllable job processing times, *AIIE Trans.* 12 (3) (1980) 258–262.
- [29] E. Nowicki, S. Zdrzałka, A two-machine flow shop scheduling problem with controllable job processing times, *European J. Oper. Res.* 34 (2) (1988) 208–220.
- [30] A. Janiak, Minimization of resource consumption under a given deadline in the two-processor flow-shop scheduling problem, *Inform. Process. Lett.* 32 (3) (1989) 101–112.
- [31] C.T.D. Ng, T.C.E. Cheng, M.Y. Kovalyov, S.S. Lam, Single machine scheduling with a variable common due date and resource-dependent processing times, *Comput. Oper. Res.* 30 (8) (2003) 1173–1185.
- [32] D. Shabtay, Single and two-resource allocation algorithms for minimizing the maximal lateness in a single machine, *Comput. Oper. Res.* 31 (8) (2004) 1303–1315.
- [33] R.L. Daniels, R.K. Sarin, Single-machine scheduling with controllable processing times and number of jobs tardy, *Oper. Res.* 37 (6) (1989) 981–984.
- [34] A. Janiak, M.Y. Kovalyov, Single machine scheduling subject to deadlines and resource dependent processing times, *European J. Oper. Res.* 94 (2) (1996) 284–291.
- [35] A. Shioura, N.V. Shakhlevich, V.A. Strusevich, Application of submodular optimization to single machine scheduling with controllable processing times subject to release dates and deadlines, *Inf. J. Comput.* 28 (1) (2016) 148–161.
- [36] C.L. Monma, A. Schrijver, M.J. Todd, V.K. Wei, Convex resource allocation problems on directed acyclic graphs: duality, complexity, special cases, and extensions, *Math. Oper. Res.* 15 (4) (1990) 736–748.
- [37] C.Y. Lee, L. Lei, Multiple-project scheduling with controllable project duration and hard resource constraint: Some solvable cases, *Ann. Oper. Res.* 102 (2001) 287–307.
- [38] M. Kaspi, D. Shabtay, A bicriterion approach to time/cost trade-offs in scheduling with convex resource-dependent job processing times and release dates, *Comput. Oper. Res.* 33 (10) (2006) 3015–3033.
- [39] L. Yedidsion, D. Shabtay, M. Kaspi, A bicriteria approach to minimize maximal lateness and resource consumption for scheduling a single machine, *J. Sched.* 10 (6) (2007) 341–352.
- [40] X. Liang, H. Chen, R. Xu, Approximately optimal algorithms for scheduling a single machine with general convex resource functions, *Int. J. Comput. Integr. Manuf.* 28 (9) (2015) 920–935.
- [41] D. Biskup, H. Jahnke, Common due date assignment for scheduling on a single machine with jointly reducible processing times, *Int. J. Prod. Econ.* 69 (3) (2001) 317–322.
- [42] M. Sklar, Fonctions de repartition n dimensions et leurs marges, Université Paris 8, 1959.
- [43] B. Schweizer, E.F. Wolff, On nonparametric measures of dependence for random variables, *Ann. Stat.* (1981) 879–885.
- [44] R.B. Nelsen, An Introduction to Copulas, Springer Science & Business Media, 2007.
- [45] S. Demarta, A.J. McNeil, The t copula and related copulas, *Int. Stat. Rev.* (2005) 111–129.
- [46] N. Whelan, Sampling from Archimedean copulas, *Quant. Finance* 4 (3) (2004) 339–352.
- [47] R. Graham, E. Lawler, J. Lenstra, A. Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Ann. Discrete Math.* 5 (2) (1979) 287–326.
- [48] J. Ceberio, E. Irurozki, A. Mendiburu, J.A. Lozano, A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems, *Progr. Artif. Intell.* 1 (2012) 103–117.
- [49] S. Baluja, Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Technical Report, Carnegie Mellon University, Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, Pittsburgh, PA, USA, 1994.
- [50] M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in: International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Springer, 1998, pp. 418–427.
- [51] P. Larrañaga, A review on estimation of distribution algorithms, in: P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Springer US, Boston, MA, 2002, pp. 57–100.
- [52] M. Pelikan, H. Muehlenbein, The bivariate marginal distribution algorithm, in: Advances in Soft Computing: Engineering Design and Manufacturing, Springer London, London, 1999, pp. 521–535.
- [53] J.S. De Bonet, C.L. Isbell, P. Viola, et al., MIMIC: Finding optima by estimating probability densities, *Adv. Neural Inf. Process. Syst.* (1997) 424–430.
- [54] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, BOA: The Bayesian optimization algorithm, in: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, vol. 1, Morgan Kaufmann Publishers Inc., 1999, pp. 525–532.
- [55] J. Ceberio, E. Irurozki, A. Mendiburu, J.A. Lozano, A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem, *IEEE Trans. Evol. Comput.* 18 (2) (2014) 286–300.
- [56] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, (2000) 201–204.
- [57] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Norwell, MA, USA, 2001.
- [58] L.-F. Wang, J.-C. Zeng, Y. Hong, Estimation of distribution algorithm based on copula theory, in: 2009 IEEE Congress on Evolutionary Computation, 2009, 1057–1063.
- [59] B. Ye, H. Gao, X. Wang, J. Zeng, Estimation of distribution algorithm based on nested Archimedean copulas constructed with Levy Subordinators, in: 2010 IEEE 11th International Conference on Computer-Aided Industrial Design Conceptual Design 1, vol. 2, 2010, pp. 1586–1590.
- [60] A. Cuesta-Infante, R. Santana, J.I. Hidalgo, C. Bielza, P.L. naga, Bivariate empirical and n-variate Archimedean copulas in estimation of distribution algorithms, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
- [61] H.D. De Mello, A.V. da Cruz, M.M. Vellasco, Estimation of distribution algorithm based on a multivariate extension of the archimedean copula, in: Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on, IEEE, 2013 pp. 75–80.
- [62] R. Salinas-Gutiérrez, A. Hernández-Aguirre, E.R. Villa-Diharce, Copula selection for graphical models in continuous estimation of distribution algorithms, *Comput. Stat.* 29 (3–4) (2014) 685–713.
- [63] Y. Gao, L. Peng, F. Li, M. Liu, X. Hu, Multiobjective estimation of distribution algorithms using multivariate archimedean copulas and average ranking, in: Foundations of Intelligent Systems, Springer, 2014, pp. 591–601.
- [64] Q. Lu, X. Yao, Clustering and learning gaussian distribution for continuous optimization, *IEEE Trans. Syst. Man Cybern. C* 35 (2) (2005) 195–204.
- [65] B. Yuan, M. Gallagher, Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA, in: 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1792–1799 Vol. 2.

- [66] W. Su, M.Y. Chow, Performance evaluation of an EDA-based large-scale plug-in hybrid electric vehicle charging algorithm, *IEEE Trans. Smart Grid* 3 (1) (2012) 308–315.
- [67] B. Schweizer, A. Sklar, *Probabilistic Metric Spaces*, Courier Corporation, 2011.
- [68] C. Alsina, B. Schweizer, M.J. Frank, *Associative Functions: Triangular Norms and Copulas*, World Scientific, 2006.
- [69] A.W. Marshall, I. Olkin, Families of multivariate distributions, *J. Amer. Statist. Assoc.* 83 (403) (1988) 834–841.
- [70] X.L. Liang, H.P. Chen, J.A. Lozano, A Boltzmann-based estimation of distribution algorithm for a general resource scheduling model, *IEEE Trans. Evol. Comput.* 19 (6) (2015) 793–806.
- [71] V. Caraffa, S. Ianes, T.P. Bagchi, C. Sriskandarajah, Minimizing makespan in a blocking flowshop using genetic algorithms, *Int. J. Prod. Econ.* 70 (2) (2001) 101–115.
- [72] E. Taillard, Benchmarks for basic scheduling problems, *Eur. J. Oper. Res.* 64 (2) (1993) 278–285.