

# Reduction of carbon emissions and project makespan by a Pareto-based estimation of distribution algorithm

Huan-yu Zheng, Ling Wang\*

Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China

## ARTICLE INFO

### Article history:

Received 28 March 2014

Accepted 6 December 2014

Available online 15 December 2014

### Keywords:

Low-carbon production

Project scheduling

Estimation of distribution algorithm

Probability model

Multi-objective optimization

## ABSTRACT

Due to the increasing concerns about global warming, low-carbon production has been a hot topic around the world. In this paper, carbon emissions reduction and project makespan minimization are considered simultaneously. To formulate the problem, a multi-objective multi-mode resource-constrained project scheduling model with makespan and carbon emissions criteria is given. To solve the problem, a Pareto-based estimation of distribution algorithm (PBEDA) is proposed. Specifically, an activity-mode list is used to encode the individual of the population; a hybrid probability model is built to describe the probability distribution of the solution space; and two Pareto archives are adopted to store the explored non-dominated solutions and the solutions for updating the probability model, respectively. New individuals are generated in the promising search areas by sampling and updating the hybrid probability model. Besides, Taguchi method of design of experiments is adopted to study the effect of parameter setting. Finally, numerical results and the comparisons to other algorithms are provided to show the effectiveness of the PBEDA in terms of quantity and quality of the obtained solutions. The Pareto set derived by the PBEDA can be helpful for project manager to recognize the relationship between carbon emissions and makespan so as to properly trade-off the two criteria according to certain preference.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Currently, global climate change is one of the world's biggest public problems. As the concentration of CO<sub>2</sub> in atmosphere is dangerously high, accompanied by more serious greenhouse effect, carbon emissions should be reduced. Although world carbon emission intensity has been reduced by 3% from 1980 to 2010, carbon emission rate has increased by 72% over the same time (Plambeck and Toktay, 2013). Facing the serious global climate change, several main economies are formulating stringent legislations or regulations on greenhouse gas emissions. Many companies are facing new constraints and trying to reduce carbon emissions while maximizing production profit.

As the largest energy-consuming industries, the industrial sector accounts for about half of world's total energy consumption (Fang et al., 2011), especially for some developing countries, e.g. more than 70% in China (Zhang and Wang, 2014). The current situation in China is "rich coal, less oil, gas shortage", i.e., coal is the main energy resource (Yu et al., 2014). Although the Chinese government and enterprises have taken great efforts, such as to shut down high energy-consumption firms, adoption of new energy-efficient technology, preference of renewable energy in the production process, carbon

emissions are still increasing rapidly as a result of economic growth. It was pointed out by Zhang and Wang (2014) that technology innovation alone is not sufficient to achieve the national carbon emission reduction target. According to Benjaafar et al. (2013), a more significant factor of carbon emissions are driven by business practices and operational policies rather than physical processes. Therefore, many measures should be taken to reduce carbon emissions by adopting advanced production, supply chain, and logistics methods.

In view of production economics, some methods to tackle carbon emissions can be considered at strategic, tactical and operational levels (Absi et al., 2013). At the strategic level, the design of supply chain or placement of factories and warehouses impacts carbon emissions (Sundarakani et al., 2010; Chaabane et al., 2012). At the tactical level, emissions are managed in production and distribution planning (Hua et al., 2011; Absi et al., 2013). At the operations level, emissions can be considered in production scheduling (Fang et al., 2011; Luo et al., 2013). At the operations level, the existing research work about low-carbon production scheduling is very limited. Fang et al. (2011) presented a multi-objective mixed integer programming model of the flow shop scheduling problem considering the reduction of three objectives: peak power load, carbon emissions and the makespan. Luo et al. (2013) proposed a new ant colony optimization for a bi-objective hybrid flow shop scheduling problem reducing makespan and electric power cost with time-of-use electricity prices. Liu (2014) developed an  $\epsilon$ -archived genetic algorithm (GA) for two batch scheduling problems

\* Corresponding author. Tel.: +86 10 62783125; fax: +86 10 62786911.

E-mail address: [wangling@tsinghua.edu.cn](mailto:wangling@tsinghua.edu.cn) (L. Wang).

to minimize CO<sub>2</sub> emissions and total weighted tardiness, while Liu and Huang (2014) studied two cases using multi-objective GA. In most of the existing literature, peak power load is regarded as an objective, while it should be kept below certain engineering limit (Kobayashi et al., 2007). Therefore, it is important to model the problem of balancing production effectiveness and carbon emissions reasonably as well as to solve the problem with novel and effective approaches.

Essentially, it is a multi-objective optimization problem considering production effectiveness and carbon emissions simultaneously. During recent years, meta-heuristics have been widely used to solve the production scheduling problems with multiple objectives, such as GA (Chang et al., 2002, 2007, 2008; Li and Wang, 2007; Chang and Chen, 2009), differential evolution (Pan et al., 2008; Qian et al., 2008, 2009), particle swarm optimization (Li et al., 2008), artificial bee colony (Wang et al., 2012b), and estimation of distribution algorithm (EDA) (L. Wang et al., 2013a, 2013b). Among them, EDA (Larrañaga and Lozano, 2002) is a population-based search framework by building a probability model for sampling new individuals among the promising areas, which is different from GA that generates new individuals by crossover and mutation operators. So far, the EDA-based algorithms have been successfully applied to a variety of optimization problems in both academic and industrial fields, such as feature selection (Armananzas et al., 2011), multi-dimensional knapsack problem (Wang et al., 2012a, 2012b), hybrid electric vehicle charging (Su and Chow, 2012), single machine scheduling (Chen et al., 2010), shop scheduling (S. Wang et al., 2013a, 2013b; Chang and Chen, 2014), and system-level synthesis (Wang et al., 2014).

For multi-objective scheduling problems (L. Wang et al., 2013a, 2013b), the EDAs have been proved to be more effective than other evolutionary algorithms. To the best of our knowledge, there is no work about EDA for solving the low-carbon production scheduling problem. In this paper, the reduction of carbon emissions and project makespan is formulated as a **Multi-Objective Multi-mode Resource-Constrained Project Scheduling Problem with Makespan and Carbon emissions** criteria (MOMRCPSP-Makespan-Carbon). A Pareto-based EDA (PBEDA) is proposed to construct a Pareto set for the problem effectively. To be specific, an activity-mode list (AML) is adopted to encode the individuals, and a modified serial schedule generation scheme (SSGS) is used to decode the AML. Moreover, a hybrid probability model is built for generating new individuals, which is updated by using the non-dominated solutions in each generation that are stored in an updating archive. Meanwhile, a Pareto archive is adopted to store all the non-dominated solutions explored during the whole search process. With the Pareto set, a proper plan can be made according to the project manager's preference. To show the effect of parameter setting, investigation by the Taguchi method of design-of-experiment method is carried out. Numerical results and the comparisons to other algorithms are provided, which demonstrate the effectiveness of the proposed PBEDA.

The remaining contents are organized as follows. In Section 2, the project scheduling model is presented to formulate the low-carbon production scheduling problem. Following the brief introduction to the original EDA in Section 3, the PBEDA to derive a Pareto set for the problem is proposed in Section 4. Numerical results and comparisons are provided in Section 5. Finally, we end the paper with some conclusions and future work in Section 6.

## 2. Project scheduling model for low-carbon production scheduling

### 2.1. Low-carbon production scheduling problem

In today's low-carbon manufacturing industry, production efficiency and carbon emissions should be considered simultaneously by project managers. The low-carbon production scheduling problem

considers a production project which consists of a series of working procedures. In each procedure, a particular amount of work should be done on machines. The goal is to reduce the project makespan and carbon emissions by reasonably choosing the processing speed and making a schedule of the procedures. Usually, it assumes that

- (1) Intermediate storage between machines is unlimited.
- (2) The actual processing speed of each machine can be chosen from a set of finite and discrete set of speeds. Under different speeds, the duration and the energy consumption of the procedure are different.
- (3) The quantity of carbon emissions per unit of consumed electricity is a constant.
- (4) A peak power load constraint should not be violated during the project execution, as mentioned in Section 1.

This work is inspired by the research and production activities in a metal-forming industry, which is an important link in manufacturing chain, supplying metal products to major manufacturing companies (Balakrishnan and Brown, 1996). Steel production includes several process phases, and is very intensive in energy consumption (Tang et al., 2002). It is a typical case of the general carbon-efficient scheduling project to make the cast iron with slots. Several activities are included in the project which can be processed in different cutting speeds, such as face milling to prepare the surface, and profile milling to cut the slots.

A typical power profile of an activity is illustrated in Fig. 1 (Fang et al., 2011). As we only consider the duration, energy assumption and peak power, the presentation of the activity can be simplified as Fig. 2. The duration of an activity  $j$  in mode  $m \in M_j$  is divided into two parts: the first part is the cutting part, where the power  $P_{jm1}$  is the peak power of the activity, and  $d_{jm1}$  is the duration of cutting process; the second part describes the rest energy assumption of the activity, where  $P_{jm2}$  is the power, and  $d_{jm2}$  is the duration of the second part. The relationship between a simplified activity and the original one is as follows:

$$P_{jm1} = P_{cutting} + P_{idle} + P_{basic} \quad (1)$$

$$d_{jm1} = t_{cutting} \quad (2)$$

$$P_{jm1}d_{jm1} + P_{jm2}d_{jm2} = P_{basic}t_{basic} + P_{idle}t_{idle} + P_{cutting}t_{cutting} \quad (3)$$

$$d_{jm2} = t_{basic} - t_{cutting} \quad (4)$$

where  $t_{basic}$ ,  $t_{idle}$ ,  $t_{cutting}$  and  $P_{basic}$ ,  $P_{idle}$ ,  $P_{cutting}$  denote the time and power in the basic, idle and cutting phases of an activity, respectively.

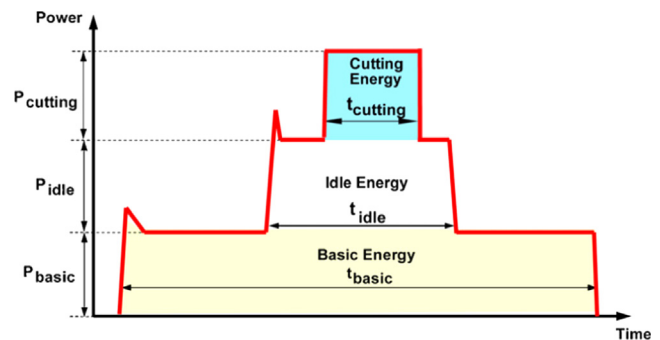


Fig. 1. Power profile for an activity (Fang et al., 2011).

## 2.2. Resource-constrained project scheduling model

The resource-constrained project scheduling problem (RCPSP) is one of the most important problems in scheduling field, where scarce resources are allocated to dependent activities over time (Brucker et al., 1999). During the past few decades, the RCPSP and its extended problems have been widely studied (Agarwal et al., 2007; Shukla et al., 2008; Fang and Wang, 2012; Wang and Fang, 2012). A majority of realistic problems have also been formulated as RCPSP, such as shop scheduling problem (Voß and Witt, 2007), service centers management (Quintanilla et al., 2012), system-level synthesis problem (Wang et al., 2014), and software project management (Alba and Francisco Chicano, 2007; Chen and Zhang, 2013).

According to the characteristics of the low-carbon production scheduling problem, it can be modeled as MOMRCPSP-Makespan-Carbon: A project contains  $J$  activities, and each activity  $j = 1, 2, \dots, J$  can be executed in  $M_j$  modes, corresponding to different processing speeds. Before activity  $j$  starts, a set of activities should be completed, which are denoted as the precedence activity set  $PS_j$ . The only renewable resource constraint is the peak power load limit  $P_{max}$ .

The start and the end of a project are presented by dummy activities 0 and  $J+1$ , respectively. The power consumption and durations of dummy activities are zero. The extended activity set  $\{0, 1, \dots, J+1\}$  is denoted as  $J^+$ . The objective of the MOMRCPSP-Makespan-Carbon is to minimize both the makespan and the carbon emissions. Thus, the mathematical model of the MOMRCPSP-Makespan-Carbon can be formulated as follows:

$$\text{Minimize}[\text{Makespan}, \text{Carbon}] \quad (5)$$

$$\text{Subject to Makespan} = \sum_{t=EFT_{J+1}}^{LFT_{J+1}} t \cdot x_{J+1,1,t} \quad (6)$$

$$E = \sum_{j=1}^J \sum_{m=1}^{M_j} (P_{jm1} \cdot d_{jm1} + P_{jm2} \cdot d_{jm2}) \cdot \sum_{t=EFT_j}^{LFT_j} x_{jmt} \quad (7)$$

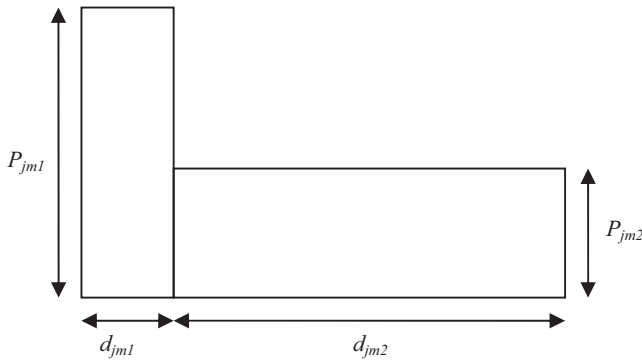


Fig. 2. A simplified activity.

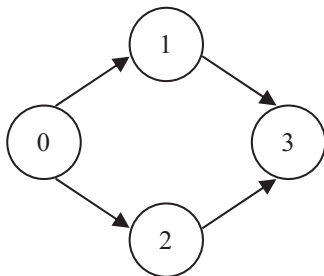


Fig. 3. An example of the MOMRCPSP-Makespan-Carbon.

$$\text{Carbon} = \varepsilon \cdot E \quad (8)$$

$$\sum_{m=1}^{M_j} \sum_{t=EFT_j}^{LFT_j} x_{jmt} = 1, \quad j \in J^+ \quad (9)$$

$$\sum_{m=1}^{M_i} \sum_{t=EFT_i}^{LFT_i} t \cdot x_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=EFT_j}^{LFT_j} (t - d_{jm1} - d_{jm2}) \cdot x_{jmt}, \quad j \in J^+, \quad i \in PS_j \quad (10)$$

$$\begin{aligned} & \sum_{j=1}^J \sum_{m=1}^{M_j} P_{jm1} \cdot \sum_{\tau=\max\{t+d_{jm1}, EFT_j\}}^{\min\{t+d_{jm1}+d_{jm2}-1, LFT_j\}} x_{jm\tau} \\ & + \sum_{j=1}^J \sum_{m=1}^{M_j} P_{jm2} \cdot \sum_{\tau=\max\{t, EFT_j\}}^{\min\{t+d_{jm1}-1, LFT_j\}} x_{jm\tau} \leq P_{max}, \quad t = 1, 2, \dots, T \end{aligned} \quad (11)$$

$$x_{jmt} = \begin{cases} 1 & \text{if activity } j \text{ is performed in mode } m \text{ and finished at time } t \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where  $x_{jmt}$  is a decision variable that represents whether activity  $j$  is executed in mode  $m$  and finishes at time  $t$ , as shown in Eq. (12). Eq. (5) defines the goal is to minimize the makespan and carbon emissions; Eqs. (6) and (7) are used to calculate the makespan and the project electric energy consumption  $E$ , respectively; Eq. (8) describes the relationship between carbon emissions and the electric energy consumption, in which  $\varepsilon$  is the quantity of carbon emissions per unit of electric energy consumption; Eq. (9) guarantees that each activity is executed only once; Eq. (10) shows the precedence constraints of the project; Eq. (11) confirms that the peak power load limit should not be exceeded;  $LFT_j$  and  $EFT_j$  are the latest finish time and the earliest finish time of activity  $j$ , respectively;  $T$  is an upper bound for the project makespan.

Let  $x = \{x_{jmt}, j \in J^+, m = 1, 2, \dots, M_j, t = 1, 2, \dots, T\}$  be a solution and  $\mathcal{X}$  be the set of all solutions. Consider two solutions  $x_1, x_2 \in \mathcal{X}$ ,  $x_1$  dominates  $x_2$  ( $x_1 \succ x_2$ ) if

$$MS(x_1) \leq MS(x_2) \quad \text{and} \quad \text{Carbon}(x_1) \leq \text{Carbon}(x_2) \quad (13)$$

where at least one of the inequalities is strict. A solution  $x$  is called a Pareto optimal solution or a non-dominated solution iff there does not exist another solution  $x' \in \mathcal{X}$  that dominates  $x$ . The Pareto set is constituted by all Pareto optimal solutions.

A simple example of the MOMRCPSP-Makespan-Carbon with four activities is shown in Fig. 3, in which activity 0 and activity 3 are dummy activities. An activity-on-node network is used to represent the project, where activities are denoted as nodes and precedent restraints between activities are denoted as arcs. Each non-dummy activity has two modes. The power requirements and corresponding durations are shown in Table 1. The peak power load limit is 16, i.e.,  $P_{max} = 16$ , and the quantity of carbon emissions per unit of electric energy consumption is 1, i.e.,  $\varepsilon = 1$ . The Pareto set is presented in Fig. 4.

## 3. Estimation of distribution algorithm

As a relatively new paradigm in the field of evolutionary computation, EDA (Larrañaga and Lozano, 2002; Lozano et al., 2006) is a general framework of statistical learning based optimization algorithm. With statistical analysis tool, the EDA makes the population movement track the promising search areas based on the available experience. For a minimization problem, the procedure of the EDA is shown below.

- Step 1: Initialize the population and the probability model.
- Step 2: Sample the probability model to generate a new population.

Step 3: Construct the elite population and update the probability model.

Step 4: Go to Step 2 unless a stopping criterion is met.

The critical step of the EDA procedure is to estimate the probability distribution. According to the characteristics of different problems, different probability models should be established to estimate the reasonable promising probability distribution. In the next section, a specially designed EDA with a probability model and an updating mechanism will be presented to obtain a Pareto set for the low-carbon production scheduling problem.

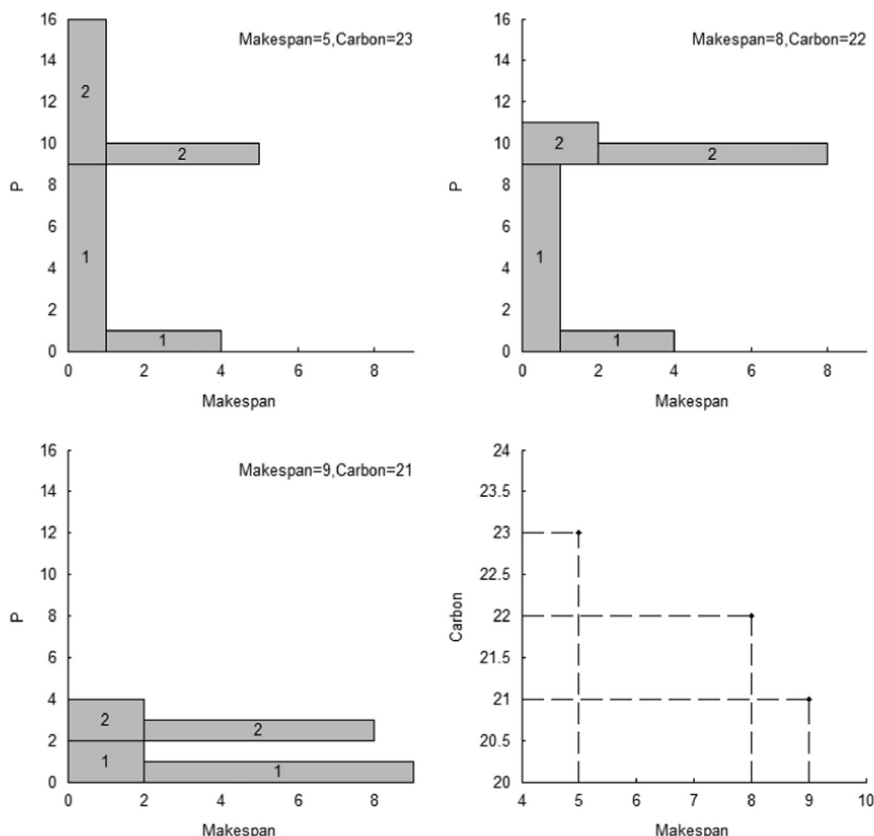
#### 4. The Pareto based EDA

In this section, the encoding and decoding schemes, initialization, hybrid probability model, sampling mechanism, and updating mechanism will be presented sequentially. Then, the framework of the PBEDA will be given and the computational complexity will be analyzed.

**Table 1**

Power requirements and corresponding durations of the example in Fig. 3.

Activity	Mode 1				Mode 2			
	$d_{jm1}$	$P_{jm1}$	$d_{jm2}$	$P_{jm2}$	$d_{jm1}$	$P_{jm1}$	$d_{jm2}$	$P_{jm2}$
1	1	9	3	1	2	2	7	1
2	1	7	4	1	2	2	6	1



**Fig. 4.** Pareto set for the example in Fig. 3.

#### 4.1. Encoding and decoding schemes

Experimental evaluation by Hartmann and Kolisch (2000) showed that metaheuristics using the activity list perform generally better than those adopting other representations. As the MOMRCSP-Makespan-Carbon is a generalization of the RCPSP, the AML based encoding scheme is adopted in the PBEDA expecting to get good results. That is,  $AML = [\pi, \lambda]$ , where  $\pi = [\pi_1, \pi_2, \dots, \pi_J]$  is the activity list, denoting the priority of activities; and  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_J]$  is the mode list, where  $\lambda_j \in \{1, 2, \dots, M_j\}$ , representing the execution mode of each activity.

Note that activities in this paper are different from the original form (see Section 2). The serial schedule generation scheme (Kolisch, 1996) (see Fig. A1 for pseudo code) is modified to build a schedule (a solution to an individual) according to activities in this paper. When an AML is given, the activities are scheduled in the activity list order as early as possible, satisfying the precedence constraints and peak power load limit. The pseudo code of the scheme is illustrated in Fig. 5, where  $n$  is the iteration;  $S_n$  denotes the set of scheduled activities;  $D_n$  is the decision set, containing all schedulable activities;  $\pi P_t$  denotes the deviation from the peak power load limit at time  $t$ ;  $A_{1t}$  and  $A_{2t}$  represent sets of activity's first and second part active at the schedule time, respectively.

#### 4.2. Initialization

The population of PBEDA is initialized by generating  $Popsiz$  individuals (decoded as AML). To be specific, an AML is generated as follows.

First, for each non-dummy activity, a possible mode is uniformly and randomly chosen to generate a mode list. Then, an iterative procedure is followed to generate the activity list. After the  $l$ th iteration, the set of selected activities are denoted as  $SA_l$ . If



**Modified SSGS**

$n = 1, S_n = \emptyset;$

**While**  $|S_n| < J$

$D_n = \{j \mid j \notin S_n, PS_j \subseteq S_n\};$

$\pi P_t = P_{\max} - \sum_{j \in A_{1t}} P_{jm_j,1} - \sum_{j \in A_{2t}} P_{jm_j,2}, t = 1, 2, \dots, T;$

$j^* = \min_{j \in D_n} \{j \mid \pi_j = \inf_{i \in D_n} \pi_i\};$

$EFT_{j^*} = \max \{FT_i \mid i \in PS_{j^*}\} + d_{j^*m_{j^*,1}} + d_{j^*m_{j^*,2}};$

$FT_{j^*} = \min \{t \mid EFT_{j^*} \leq t \leq LFT_{j^*},$

$P_{1j^*} \leq \pi P_{\tau_1}, \tau_1 = t - d_{j^*m_{j^*,1}} - d_{j^*m_{j^*,2}} + 1, \dots, t - d_{j^*m_{j^*,2}},$

$P_{2j^*} \leq \pi P_{\tau_2}, \tau_2 = t - d_{j^*m_{j^*,2}} + 1, \dots, t\};$

$S_{n+1} = S_n \cup \{j^*\};$

$n = n + 1;$

**End while**

**Fig. 5.** Pseudo code of the modified SSGS.

$l = 1, SA_l = \{0\}$ , which means only the dummy activity 0 is selected in the first iteration. Then, an eligible set  $SE_l = \{j \mid j \in J^+ \setminus SA_l, PS_j \subseteq SA_l\}$  is determined to define the activities that can start but are not selected. The probability of activity  $j \in SE_l$  adding to  $SA_l$  is  $\psi_j$ , following the latest-finish-time (LFT) priority rule (Davis and Patterson, 1975) as follows:

$$\psi_j = (v_j + 1) / \sum_{i \in SE_l} (v_i + 1) \quad (14)$$

$$v_j = \max_{i \in SE_l} LFT_i - LFT_j \quad (15)$$

Then,  $l = l + 1$ , and it enters the next iteration. The process repeats until all activities are selected into the activity list. This method is also known as the regret-based biased random sample method (Kolisch and Drexel, 1996).

#### 4.3. Hybrid probability model

It is crucial to design the probability model for EDA (Lozano et al., 2006). Here, a hybrid probability model is presented to predict and track the promising search areas of the MOMRCPSP-Makespan-Carbon. The hybrid probability model contains two probability matrixes: **PMAct** and **PMMMod**.

- (1) Probability matrix of activity (**PMAct**) is used to predict the positions of activities in the activity list.

$$\mathbf{PMAct}(g) = \begin{bmatrix} \alpha_{11}(g) & \cdots & \alpha_{1J}(g) \\ \vdots & \ddots & \vdots \\ \alpha_{J1}(g) & \cdots & \alpha_{JJ}(g) \end{bmatrix} \quad (16)$$

where  $\alpha_{ji}(g)$  is the probability of placing activity  $j$  at the  $i$ th position in the  $g$ th generation. The probability matrix is uniformly and randomly initialized as follows:

$$\mathbf{PMAct}(0) = \begin{bmatrix} 1/J & \cdots & 1/J \\ \vdots & \ddots & \vdots \\ 1/J & \cdots & 1/J \end{bmatrix} \quad (17)$$

- (2) Probability matrix of mode (**PMMMod**) is used to predict the execution mode of activities in the mode list.

$$\mathbf{PMMMod}(g) = \begin{bmatrix} \mu_{11}(g) & \cdots & \mu_{1M}(g) \\ \vdots & \ddots & \vdots \\ \mu_{J1}(g) & \cdots & \mu_{JM}(g) \end{bmatrix} \quad (18)$$

where  $M = \max_{j=1,2,\dots,J} M_j$ , and  $\mu_{ji}(g)$  is the probability of activity  $j$  executed with mode  $i$  at generate  $g$ . The probability matrix is uniformly initialized as follows, to explore the whole search space at the beginning of the algorithm.

$$\mathbf{PMMMod}(0) = \begin{bmatrix} \delta_{11}/M_1 & \cdots & \delta_{1M}/M_1 \\ \vdots & \ddots & \vdots \\ \delta_{J1}/M_J & \cdots & \delta_{JM}/M_J \end{bmatrix} \quad (19)$$

$$\delta_{ji} = \begin{cases} 1 & \text{if } i \leq M_j, \\ 0 & \text{else} \end{cases}, \quad 1 \leq i \leq M, \quad 1 \leq j \leq J. \quad (20)$$

#### 4.4. Probability-model-based sampling mechanism

In each generation, the individuals are generated by sampling AMLs using the hybrid probability model. Specifically, for activity list, the selection probability of activity  $j$  at each position  $i$  ( $Pa_{ji}$ ) is calculated according to **PMAct** over the eligible activity set **SE** as follows:

$$Pa_{ji} = \alpha_{ji} / \sum_{h \in SE} \alpha_{hi} \quad (21)$$

If activity  $j$  has been selected into the activity list, all elements in row  $j$  of **PMAct** ( $\alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jJ}$ ) are set to zero. As a result, each activity can be selected only once in the activity list. After selecting activity one by one, the activity list is formed.

Similarly, the mode list can be obtained by sampling **PMMMod**. An execution mode  $i$  is assigned to each activity  $j$  in the activity list with probability  $Pm_{ji}$  as Eq. (22). Then, the modified SSGS is used to transform the generated AML to a schedule.

$$Pm_{ji} = \mu_{ji} / \sum_{h=1}^{M_j} \mu_{jh}. \quad (22)$$

#### 4.5. Pareto archive and updating archive

In the PBEDA, a Pareto archive **PA** is used to store the non-dominated solutions that have been explored (say approximate Pareto set), and an updating archive **UA** is used to store the newly found non-dominated solutions in each generation. The solution  $SN_i$  ( $i = 1, 2, \dots, \text{Popsiz}$ ) related to each newly generated individual is compared with each solution  $SP_j$  ( $j = 1, 2, \dots, |\mathbf{PA}|$ ) in **PA**. Then, those non-dominated solutions are added to **PA** and **UA**. At the same time, solutions dominated by new found solutions are removed from **PA**. The pseudo code of the archive updating procedure is illustrated in Fig. 6.

#### 4.6. Updating mechanism

The updating archive **UA** is used to update the hybrid probability model in each generation as follows:

$$\alpha_{ji}(g+1) = (1 - \theta_1) \cdot \alpha_{ji}(g) + \frac{\theta_1}{|\mathbf{UA}|_{k \in \mathbf{UA}}} \sum I_{ji}^k, \quad 1 \leq i, \quad j \leq J \quad (23)$$

$$\mu_{ji}(g+1) = (1 - \theta_2) \cdot \mu_{ji}(g) + \frac{\theta_2}{|\mathbf{UA}|_{k \in \mathbf{UA}}} \sum R_{ji}^k, \quad 1 \leq i \leq M, \quad 1 \leq j \leq J \quad (24)$$

where  $\theta_1$  and  $\theta_2$  are the learning rates of **PMAct** and **PMMMod**, respectively;  $I_{ji}^k$  and  $R_{ji}^k$  are indicator functions for the  $k$ th individual of **UA** as Eqs. (25) and (26). Note that the hybrid probability model will not be updated if **UA** =  $\emptyset$ .

$$I_{ji}^k = \begin{cases} 1 & \text{if activity } j \text{ is placed at position } i \\ 0 & \text{else} \end{cases} \quad (25)$$

### Archive Updating

```

UA = ∅;
For  $i = 1$  to  $Popsiz$ 
    flag = true;
    For  $j = 1$  to  $|PA|$ 
        If  $SP_j \succ SN_i$  or  $SP_j = SN_i$ 
            flag = false;
            break;
        End if
    End for
    If flag == true
        For  $j = 1$  to  $|PA|$ 
            If  $SN_i \succ SP_j$ 
                Remove  $SP_j$  from PA;
            End if
        End for
        Add  $SN_i$  to PA and UA;
    End if
End for

```

Fig. 6. Pseudo code of the archive updating procedure.

$$R_{ji}^k = \begin{cases} 1 & \text{if activity } j \text{ is executed in mode } i \\ 0 & \text{else} \end{cases} \quad (26)$$

#### 4.7. Procedure of the proposed PBEDA

The procedure of the proposed PBEDA is summarized as follows. First, the population and the probability matrix **PMAct**(0) and **PMMMod**(0) are initialized. Then, a new population is generated by sampling the probability model with the designed sampling mechanism. After all the individuals are evaluated by the modified SSGS, the Pareto archive **PA** is updated, and solutions improving **PA** are selected into **UA**. With **UA**, the probability matrix **PMAct**(g) and **PMMMod**(g) are updated. The procedure is repeated until a stopping criterion is met. The flowchart of the procedure is illustrated in Fig. 7.

#### 4.8. Computational complexity analysis

For each generation of the PBEDA, its computational complexity is analyzed as follows.

For the sampling process, every position of activity list is generated by randomly sampling **PMAct**, the construction of the activity list is with the complexity  $O(j^2)$ ; every execution mode of the mode list is sampled with the **PMMMod**, the computational complexity of generating a mode list is  $O(Mj)$ . Note that the size of population is  $Popsiz$ , a population can be generated with the complexity  $O[j \cdot Popsiz \cdot (j + M)]$ .

For the decoding process, Ballestín (2007) pointed out that the complexity of the original SSGS to generate a schedule is  $O(K \cdot j^2)$ , in which  $K$  is number of renewable resources. As the peak power load limit is the only resource constraint, i.e.,  $K = 1$ , then the modified SSGS to generate a schedule is with the complexity of  $O(j^2)$ . Thus, the complexity of generating schedules for a population is  $O(Popsiz \cdot j^2)$ .

For the updating process, to update **PA** and construct **UA**, the solutions to newly generated individuals are compared to solutions

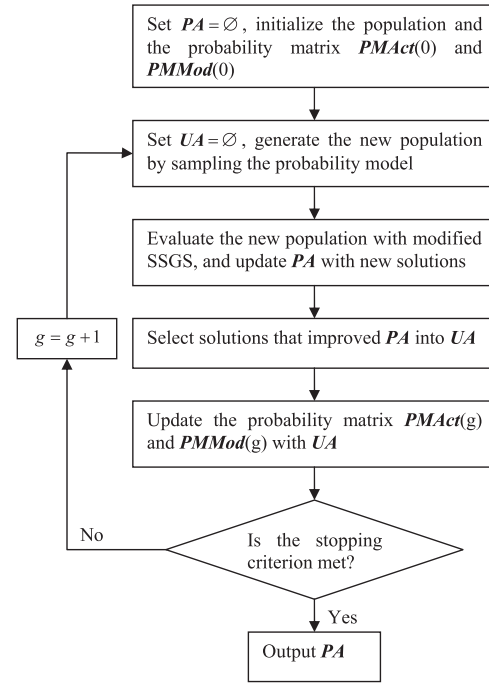


Fig. 7. The flowchart of the PBEDA.

in **PA**, with the complexity of  $O(Popsiz \cdot |PA|)$ . Usually,  $|PA|$  is similar to or less than  $J$ . Thus, the complexity of updating process can be roughly estimated as  $O(Popsiz \cdot J)$ . To update all  $j^2$  elements of **PMAct** and  $M \cdot J$  elements of **PMMMod**, it is with the complexity of  $O(j^2 + MJ)$ . So, the complexity of the updating process is  $O[j \cdot (Popsiz + J + M)]$ .

Thus, the computational complexity of each generation of the PBEDA is  $O[j \cdot Popsiz \cdot (J + M)] + O(Popsiz \cdot j^2) + O[j \cdot (Popsiz + J + M)]$ , which can be simplified as  $O[Popsiz \cdot J \cdot (2J + M)]$ . Taking the number of generation  $g$  into account, the total computational complexity of the PBEDA is  $O[g \cdot Popsiz \cdot J \cdot (2J + M)]$ . It can be seen that the complexity of the proposed PBEDA does not increase greatly compared to solution space as the size of problem increases.

## 5. Numerical results and comparisons

To test the performance of the proposed PBEDA, numerical simulations are carried out and comparisons to random search algorithm (RAND) and the well-known non-dominated sorting genetic algorithm II (NSGA2) (Deb et al., 2002) are performed. All the experiments are implemented on a PC with 2.83 GHz processor/4.00 GB RAM in Windows 7. The algorithm is coded in C++ using Microsoft Visual Studio 2008.

### 5.1. Test data description

To evaluate the performance of different algorithms, test data is needed for different situations in accordance with problem size and structure. Due to the difficulty in accessing real data from industry, a set of test data is generated randomly as follows.

First, the number of activities and the precedence relationship between activities of projects are the same as the standard multi-mode RCPSP test data sets from the well-known PSPLIB (Kolisch and Sprecher, 1996) j10, j20 and j30, which contain 536, 554, 552 projects, respectively. Then, the levels and ranges of other factors are shown in Table 2. Note that each non-dummy activity in a project is assigned three modes, corresponding to different cutting speeds. In addition, factors  $P_{cutting}$  and  $t_{cutting}$  of an activity are calculated as Eqs. (27) and (28). The quantity of carbon emissions

**Table 2**  
Factors and their levels.

Notations	Factors	Levels	Number of levels
$P_{max}$	Peak power load of a project	Discrete uniform [225, 300]	1
$U$	Amount of work of an activity	Discrete uniform [10, 50]	1
$V_{cutting}$	Cutting speed of an activity	Discrete uniform [1, 15]	3
$P_{idle}$	Idle power of an activity	Discrete uniform [2, 8]	1
$t_{idle}$	Idle time of an activity	Discrete uniform [ $t_{cutting}$ , $2t_{cutting}$ ]	1
$P_{basic}$	Basic power of an activity	Discrete uniform [1, 4]	1
$t_{basic}$	Basic time of an activity	Discrete uniform [ $2t_{idle}$ , $4t_{idle}$ ]	1
$\delta$	Ratio of $P_{cutting}$ to $V_{cutting}^2$	Continuous uniform [0, 1]	1

**Table 3**  
Combinations of parameter values.

Parameters	Factor level			
	1	2	3	4
$Popsiz$	80	100	120	140
$\theta_1$	0.01	0.05	0.1	0.5
$\theta_2$	0.01	0.05	0.1	0.5

per unit of electric energy consumption is set to 1, i.e.,  $\varepsilon = 1$ .

$$P_{cutting} = \delta V_{cutting}^2 \quad (27)$$

$$t_{cutting} = U/V_{cutting}. \quad (28)$$

## 5.2. Parameters setting

The PBEDA contains three key parameters: the population size of each generation ( $Popsiz$ ) and the learning rates of **PMAct** and **PMMod** ( $\theta_1$  and  $\theta_2$ ). The Taguchi method of design of experiment (DOE) (Montgomery, 2005) is used to investigate the effect of parameter setting on the PBEDA. Combinations of different values for these parameters are listed in Table 3. The data set with moderate scale j20 is chosen to carry out the DOE test. The stopping criterion is set to a maximum 50,000 number of schedules for each project.

According to the number of parameters and the number of factor levels, we choose the orthogonal array  $L_{16}(4^3)$ . That is, the total number of treatments is 16, the number of parameters is three and the number of factor levels is four. For the  $i$ th parameter combination,  $i = 1, 2, \dots, 16$ , the PBEDA is run to obtain a set of non-dominated solutions  $PA_j^i$  for project  $j$  of j20. The non-dominated solutions among all  $PA_j^i$  for project  $j$  are collected as the reference set  $RS_j$ . Clearly, the more solutions in  $RS_j$  obtained by one parameter combination, the better the combination is. The average response variable (ARV) is the average percentage of non-dominated solutions for each combination of  $RS_j$ , which can be calculated as Eq. (29). The orthogonal array and the obtained ARV values are shown in Table B1.

$$ARV(i) = \frac{1}{554} \sum_{j=1}^{554} \frac{|RS_j \cap PA_j^i|}{|RS_j|} \times 100\% \quad (29)$$

The trend of each factor level is shown Fig. 8. Then, the response value of each parameter is figured out and the significance rank of each parameter is analyzed in Table 4.

From Table 4, it can be seen that learning rate of **PMMod**  $\theta_2$  is the most significant parameter among the three factors. As the carbon emissions are only decided by mode list, while the make-span is decided by both activity list and mode list, the adjustment of mode list is more crucial to the performance of PBEDA. Thus,  $\theta_2$  is a more significant parameter than learning rate of **PMAct**  $\theta_1$ . From Fig. 8, it can be seen that  $\theta_2$  should neither be too small nor too large, as a large  $\theta_2$  makes the algorithm easily be trapped in local minima, while a small  $\theta_2$  leads to a slow convergence. The parameter  $\theta_1$  ranks second among the three. Similar to  $\theta_2$ , a large value of  $\theta_1$  leads to premature convergence, but it is indicated that a small  $\theta_1$  is helpful to maintain population diversity. Although the population size  $Popsiz$  has the least significance to the algorithm, a suitable one can still yield a good performance. According to the analysis above, an appropriate combination of parameters is suggested as  $Popsiz = 80$ ,  $\theta_1 = 0.01$  and  $\theta_2 = 0.05$ .

## 5.3. Pareto sets with different computing budget

In this section, the Pareto sets with different computing budget are investigated using the instance j3010\_9. The stopping criteria are set to 1000, 5000 and 50,000 schedules.

The Pareto sets of the instance j3010\_9 with different stopping criteria are shown in Fig. 9. It can be seen that as more schedules are generated, the solutions in the approximate Pareto set become better and the size of approximate Pareto set becomes larger. This implies that the PBEDA can keep on finding better and more solutions with the increase of computational budget.

## 5.4. Comparison of different algorithms

Next, the PBEDA is compared with RAND and the well-known multi-objective optimization algorithm NSGA2 (Deb et al., 2002), which adopt the same encoding and decoding procedure as PBEDA. In addition, for NSGA2, some widely used genetic operators for the RCSP are adopted: two-point crossover (Hartmann, 1998) and permutation-based local search (PBL) (Fang and Wang, 2012). To compare the performance of different algorithms, the following performance metrics are adopted:

- (1) The size of approximate Pareto set. An approximate Pareto set is better, if it contains more solutions. It means more choices can be provided to the project manager.
- (2) The coverage of the approximate Pareto sets (Zitzler and Thiele, 1999). Suppose two approximate Pareto sets  $PA_1$  and  $PA_2$  obtained by two algorithms. The coverage of  $PA_1$  and  $PA_2$ , denoted as  $C(PA_1, PA_2) \in [0, 1]$ , which is also called C values between  $PA_1$  and  $PA_2$ , is defined as follows:

$$C(PA_1, PA_2) = |\{x_2 | x_2 \in PA_2, \exists x_1 \in PA_1 : x_1 \succ x_2 \text{ or } x_1 = x_2\}| / |PA_2| \quad (30)$$

The larger value of  $C(PA_1, PA_2)$ , the more solutions that in  $PA_2$  are dominated by or equal to solutions in  $PA_1$ . If  $C(PA_1, PA_2)$  is larger than  $C(PA_2, PA_1)$ , then  $PA_1$  is better than  $PA_2$  in sense of the Pareto dominance. That is, the algorithm obtaining  $PA_1$  is better than that obtaining  $PA_2$ .

In our experiments, the population size of NSGA2 is set to 100, and the probability of exchanging two consecutive activities by PBL is set to 0.2. The stopping criteria are set to 1000, 5000 and 50,000 schedules for each project. The average sizes of the approximate Pareto sets for j10, j20 and j30 are listed in Tables 5, 6 and 7, respectively.

It can be seen from the tables that the PBEDA can provide more solutions in the approximate Pareto set than RAND and NSGA2 on

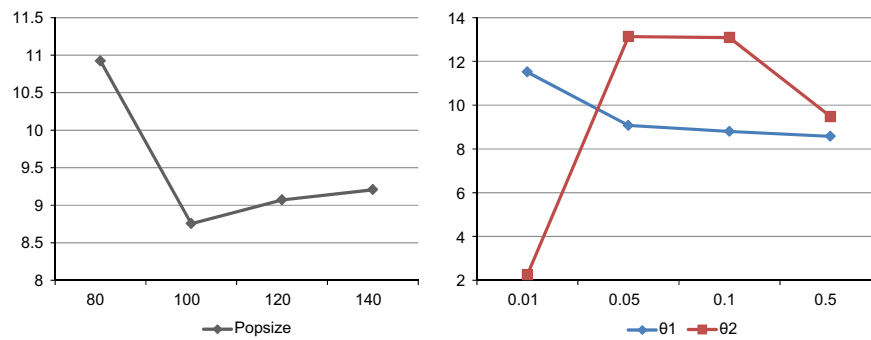


Fig. 8. Factor level trend of the PBEDA.

**Table 4**  
Response value and significance rank.

Level	Popsiz	$\theta_1$	$\theta_2$
1	10.921	11.507	2.256
2	8.755	9.070	13.132
3	9.070	8.800	13.087
4	9.206	8.574	9.477
Delta	2.166	2.933	10.875
Rank	3	2	1

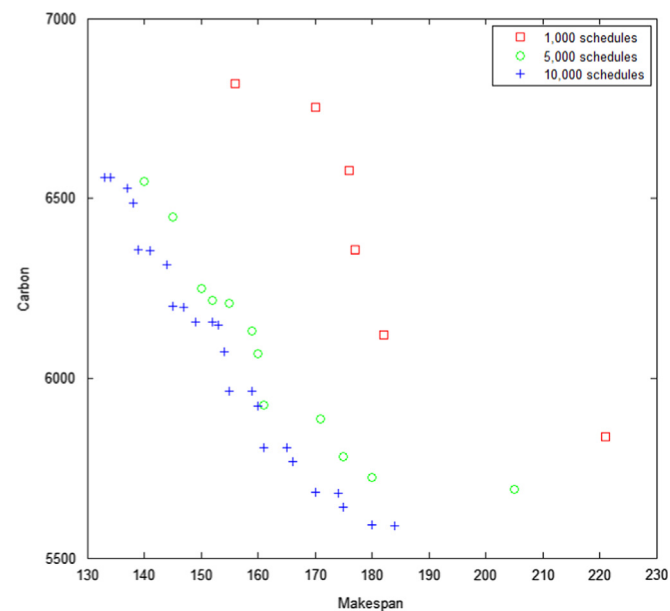


Fig. 9. Pareto front with different computing budget.

**Table 5**  
Average size of the approximate Pareto set for j10.

Algorithms	1000	5000	50,000
RAND	6.11	7.07	7.51
PBEDA	<b>6.36</b>	<b>7.46</b>	<b>7.54</b>
NSGA2	5.90	6.38	6.44

Note: the bold values mean the best results.

all cases, especially for middle to large scale problems. This implies that the PBEDA is superior to RAND and NSGA2 in terms of population diversity.

In terms of  $C$  values, the box plot is used to show the distributions intuitively. For each project in j10, j20 and j30, RAND, PBEDA and NSGA2 are compared to each other in pairs to calculate  $C$  values. The

**Table 6**  
Average size of the approximate Pareto set for j20.

Algorithms	1000	5000	50,000
RAND	5.62	7.27	9.76
PBEDA	<b>6.00</b>	<b>11.01</b>	<b>15.27</b>
NSGA2	5.95	8.12	9.28

Note: the bold values mean the best results.

**Table 7**  
Average size of the approximate Pareto set for j30.

Algorithms	1000	5000	50,000
RAND	5.21	6.62	8.89
PBEDA	<b>5.51</b>	<b>9.94</b>	<b>20.56</b>
NSGA2	5.47	7.35	11.14

Note: the bold values mean the best results.

box plots with different stopping criteria for j10, j20 and j30 are shown in Figs. 10, 11 and 12, respectively. Each subfigure contains three box plots representing the distributions of  $C$  values for pairs of algorithms when stopping criterion is set to 1000, 5000, and 50,000 schedules, respectively. The median is indicated by a line inside the box. The lower quartile  $q_1$  and upper quartile  $q_3$  are indicated by the lower and upper ends of the box (called inner fences). The lower and upper ends of whiskers indicate the outer fences of the box, which are  $4.5q_1 - 1.5q_3$  and  $4.5q_3 - 1.5q_1$ . Outliers plotted in dots indicate data that are not between the outer fences.

From Figs. 10 to 12, it can be seen that the PBEDA is superior to the NSGA2 and the RAND on all data sets with all the three stopping criteria. Taking j30 with 1000 schedules as an example, it can be seen from Fig. 12 that the median value of  $C(PBEDA, NSGA2)$  is 0.6, and the median value of  $C(NSGA2, PBEDA)$  is 0.2. Also, it can be seen from Figs. 10 to 12 that with 5000 or 50,000 generated schedules, all median values of  $C(PBEDA, NSGA2)$  and  $C(PBEDA, RAND)$  equal to 1. That is, the PBEDA is better than the other two algorithms on coverage of the approximate Pareto set especially when computational budget is medium to large.

In addition, the average CPU times spent by PBEDA, NSGA2 and RAND for each instance in j10, j20 and j30, are shown in Table 8, with maximum 50,000 schedules as the stopping criterion.

From Table 8, it can be seen that in each problem set, CPU time spent by the PBEDA is slightly more than that of the RAND and the NSGA2. The reason is that the RAND and the NSGA2 perform search on the solution space directly, while the PBEDA needs to update and sample the probability distribution to perform search process. Fortunately, the average CPU time of the PBEDA is at the same level of the RAND and the NSGA2. Above all, the Pareto set found by the PBEDA has much better quality than



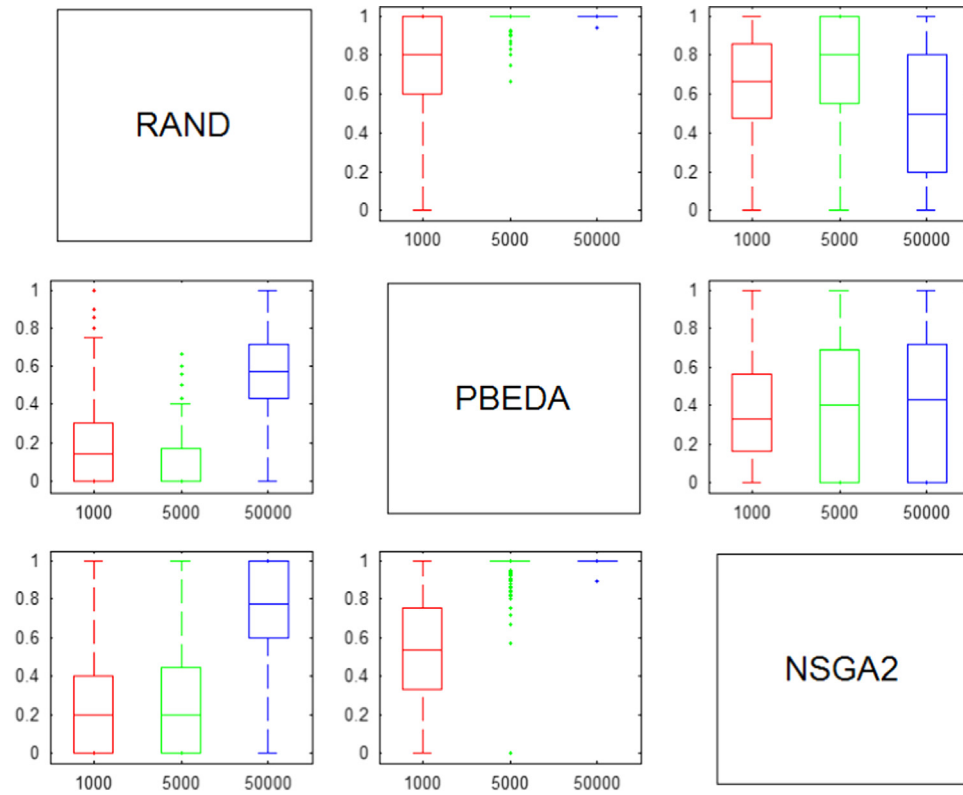


Fig. 10. Box plots of C values for j10.

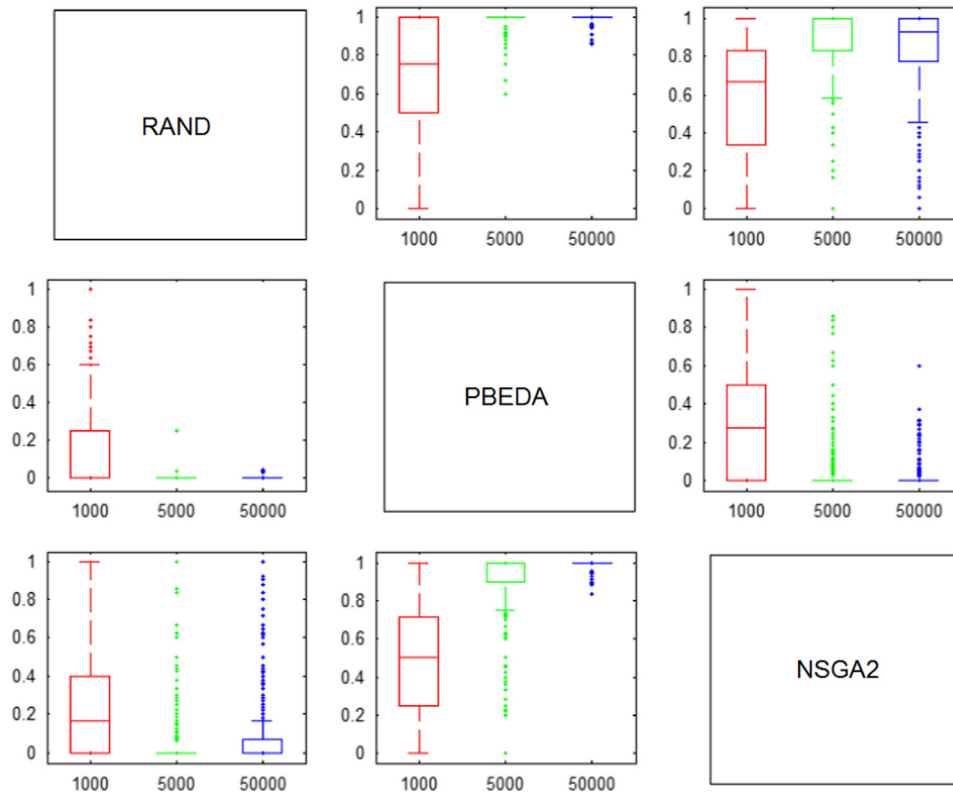


Fig. 11. Box plots of C values for j20.

those by the RAND and the NSGA2. Therefore, the PBEDA is more effective than both the NSGA2 and the RAND considering two performance metrics in terms of quantity and quality of the obtained solutions.

The effectiveness of the PBEDA owes two main reasons. First, probability model of the PBEDA can predict the movements of population in the search space and estimate the potential probability distribution of the encoded representation, so as to generate new

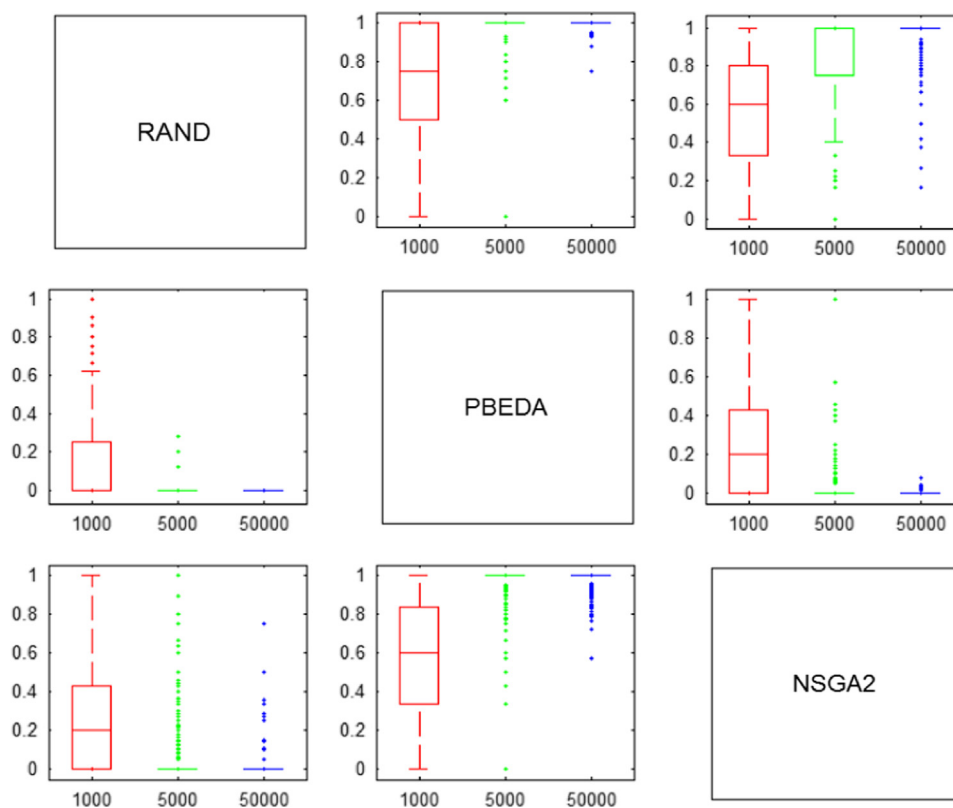


Fig. 12. Box plots of C values for j30.

**Table 8**  
Average CPU times (s).

Problem set	j10	j20	j30
RAND	0.339	1.045	2.239
PBEDA	0.444	1.601	3.754
NSGA2	0.419	1.048	2.575

individuals according to the promising search areas. Second, by using updating archive, the probability model is updated by excellent individuals, which improves the convergence rate of the probability model. The hybrid probability model of the PBEDA can provide the manager information about the distribution of project activities and modes. Then activities can be put on proper positions with proper modes. Since the PBEDA optimizes carbon reduction and project makespan at the same time, it helps the manager to understand the relationship between these two objectives. Because the PBEDA can find a better Pareto set than the RAND and the NSGA2 in both quantity and quality, more and better choices for project manager can be chosen to make a flexible decision according to certain preference between reduction of carbon emissions and production effectiveness.

## 6. Conclusions

In this paper, the low-carbon production scheduling problem was modeled as a multi-objective multi-mode resource-constrained project scheduling problem concerning both the makespan and carbon emissions criteria. Then, a Pareto-based estimation of distribution algorithm was proposed to solve the problem. Numerical results and comparisons showed the effectiveness of the algorithm in solving the low-carbon production scheduling problem. This work is the first to formulate the low-carbon production

scheduling problem as a project scheduling model, and it is also the first work to solve the problem by developing a multi-objective estimation of distribution algorithm. As the majority of the project scheduling approaches have focused on production effectiveness but ignored environmental objectives, this work is an important addition to the literature. And, this work can be useful for project manager in production industry to balance production effectiveness and carbon emissions under peak power load constraint. Future work could focus on further studying the low-carbon production issues, such as exploring problem-specific methodologies. It is also interesting to develop other novel evolutionary algorithms for the problems and to extend research on low-carbon scheduling in distributed production environments.

## Acknowledgments

This research is partially supported by the National Key Basic Research and Development Program of China (2013CB329503), the National Natural Science Foundation of China (61174189, 61025018), and the Doctoral Program Foundation of Institutions of Higher Education of China (20130002110057).

## Appendix A

See Fig. A1, where  $J$ ,  $T$  and  $PS_j$  are consistent with the notations in Section 2;  $n$ ,  $S_n$ ,  $D_n$  and  $\pi$  are consistent with the notations in Section 4.1;  $K$  represents the number of renewable resources in the project;  $\pi K_{rt}$  denotes the left over amount of renewable resource  $r$  at time period  $t$ ; for each period, there are  $K_r$  units of renewable resource  $r$ ; activity  $j$  requires  $k_{jr}$  units of renewable resource  $r$  during its duration.

**Original SSGS**

$n = 1, S_n = \emptyset$ ;

**While**  $|S_n| < J$

$D_n = \{j \mid j \notin S_n, PS_j \subseteq S_n\}$ ;

$\pi K_{rt} = K_r - \sum_{j \in A_t} k_{jr}, t = 1, 2, \dots, T, r = 1, 2, \dots, K$ ;

$j^* = \min_{j \in D_n} \{j \mid \pi_j = \inf_{i \in D_n} \pi_i\}$ ;

$EFT_{j^*} = \max\{FT_i \mid i \in PS_{j^*}\} + d_{j^*}$ ;

$FT_{j^*} = \min\{t \mid EFT_{j^*} \leq t \leq LFT_{j^*},$   
 $k_{j^*r} \leq \pi K_{rt}, \tau = t - d_{j^*} + 1, \dots, t, r \in R\}$ ;

$S_{n+1} = S_n \cup \{j^*\}$ ;

$n = n + 1$ ;

**End while**

**Fig. A1.** Pseudo code of the original SSGS (Kolisch, 1996).

**Appendix B**

See Table B1.

**Table B1**

Orthogonal array and ARV values.

Combination number $i$	Factor			ARV (%)
	Popsize	$\theta_1$	$\theta_2$	
1	1	1	1	8.48
2	1	2	2	13.18
3	1	3	3	13.36
4	1	4	4	8.66
5	2	1	2	13.54
6	2	2	1	0.18
7	2	3	4	8.48
8	2	4	3	12.82
9	3	1	3	13.36
10	3	2	4	10.11
11	3	3	1	0.18
12	3	4	2	12.64
13	4	1	4	10.65
14	4	2	3	12.82
15	4	3	2	13.18
16	4	4	1	0.18

**References**

- Absi, N., Dauzère-Pérès, S., Kedad-Sidhoum, S., Penz, B., Rapine, C., 2013. Lot sizing with carbon emission constraints. *Eur. J. Oper. Res.* 227 (1), 55–61.
- Agarwal, R., Tiwari, M.K., Mukherjee, S.K., 2007. Artificial immune system based approach for solving resource constraint project scheduling problem. *Int. J. Adv. Manuf. Technol.* 34 (5–6), 584–593.
- Alba, E., Francisco Chicano, J., 2007. Software project management with GAs. *Inf. Sci.* 177 (11), 2380–2401.
- Armananzas, R., Saey, Y., Inza, I., Garcia-Torres, M., Bielza, C., Van de Peer, Y., Larranaga, P., 2011. Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms. *IEEE-ACM Trans. Comput. Biol. Bioinform.* 8 (3), 760–774.
- Balakrishnan, A., Brown, S., 1996. Process planning for aluminum tubes: an engineering-operations perspective. *Oper. Res.* 44 (1), 7–20.
- Ballestin, F., 2007. When it is worthwhile to work with the stochastic RCPSPP? *J. Sched.* 10 (3), 153–166.
- Benjaafar, S., Li, Y., Daskin, M., 2013. Carbon footprint and the management of supply chains: insights from simple models. *IEEE Trans. Autom. Sci. Eng.* 10 (1), 99–116.

- Brucker, P., Drexel, A., Möhring, R., Neumann, K., Pesch, E., 1999. Resource-constrained project scheduling: notation, classification, models, and methods. *Eur. J. Oper. Res.* 112 (1), 3–41.
- Chaabane, A., Ramudhin, A., Paquet, M., 2012. Design of sustainable supply chains under the emission trading scheme. *Int. J. Prod. Econ.* 135 (1), 37–49.
- Chang, P.C., Chen, M.H., 2014. A block based estimation of distribution algorithm using bivariate model for scheduling problems. *Soft Comput.* 18 (6), 1177–1188.
- Chang, P.C., Chen, S.H., 2009. The development of a sub-population genetic algorithm II (SPGA II) for multi-objective combinatorial problems. *Appl. Soft Comput.* 9 (1), 173–181.
- Chang, P.C., Chen, S.H., Fan, C.Y., Chan, C.L., 2008. Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems. *Appl. Math. Comput.* 205 (2), 550–561.
- Chang, P.C., Chen, S.H., Liu, C.H., 2007. Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems. *Expert Syst. Appl.* 33 (3), 762–771.
- Chang, P.C., Hsieh, J.C., Lin, S.G., 2002. The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem. *Int. J. Prod. Econ.* 79 (3), 171–183.
- Chen, S.H., Chen, M.C., Chang, P.C., Zhang, Q., Chen, Y.M., 2010. Guidelines for developing effective estimation of distribution algorithms in solving single machine scheduling problems. *Expert Syst. Appl.* 37 (9), 6441–6451.
- Chen, W.N., Zhang, J., 2013. Ant colony optimization for software project scheduling and staffing with an event-based scheduler. *IEEE Trans. Softw. Eng.* 39 (1), 1–17.
- Davis, E.W., Patterson, J.H., 1975. A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Manag. Sci.* 21 (8), 944–955.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6 (2), 182–197.
- Fang, C., Wang, L., 2012. An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Comput. Oper. Res.* 39 (5), 890–901.
- Fang, K., Uhan, N., Zhao, F., Sutherland, J.W., 2011. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J. Manuf. Syst.* 30 (4), 234–240.
- Hartmann, S., 1998. A competitive genetic algorithm for resource-constrained project scheduling. *Nav. Res. Logist.* 45 (7), 733–750.
- Hartmann, S., Kolisch, R., 2000. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* 127 (2), 394–407.
- Hua, G., Cheng, T.C.E., Wang, S., 2011. Managing carbon footprints in inventory management. *Int. J. Prod. Econ.* 132 (2), 178–185.
- Kobayashi, M., Feng, Y., Loarte, A., Federici, G., Strohmayr, G., Shimada, M., Sardei, F., Reiter, D., Sugihara, M., 2007. 3D edge transport analysis of ITER start-up configuration for limiter power load assessment. *Nucl. Fusion* 47 (2), 61.
- Kolisch, R., 1996. Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. *Eur. J. Oper. Res.* 90 (2), 320–333.
- Kolisch, R., Drexel, A., 1996. Adaptive search for solving hard project scheduling problems. *Nav. Res. Logist.* 43 (1), 23–40.
- Kolisch, R., Sprecher, A., 1996. PSPLIB – a project scheduling problem library: OR software-ORSEP operations research software exchange program. *Eur. J. Oper. Res.* 96 (1), 205–216.
- Larrañaga, P., Lozano, J.A., 2002. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Springer, Netherlands.
- Li, B.B., Wang, L., 2007. A hybrid quantum-inspired genetic algorithm for multi-objective flow shop scheduling. *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* 37 (3), 576–591.
- Li, B.B., Wang, L., Liu, B., 2008. An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling. *IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum.* 38 (4), 818–831.
- Liu, C.H., 2014. Approximate trade-off between minimisation of total weighted tardiness and minimisation of carbon dioxide (CO<sub>2</sub>) emissions in bi-criteria batch scheduling problem. *Int. J. Comput. Integr. Manuf.* 27 (8), 759–771.
- Liu, C.H., Huang, D.H., 2014. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *Int. J. Prod. Res.* 52 (2), 337–352.
- Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E., 2006. Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Springer, New York.
- Luo, H., Du, B., Huang, G.Q., Chen, H., Li, X., 2013. Hybrid flow shop scheduling considering machine electricity consumption cost. *Int. J. Prod. Res.* 146 (2), 423–439.
- Montgomery, D.C., 2005. Design and Analysis of Experiments. John Wiley and Sons, Arizona.
- Pan, Q.K., Wang, L., Qian, B., 2008. A novel multi-objective particle swarm optimization algorithm for no-wait flow shop scheduling problems. *Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf.* 222 (4), 519–539.
- Plambeck, E.L., Toktay, L.B., 2013. Introduction to the special issue on the environment. *Manuf. Serv. Oper. Manag.* 15 (4), 523–526.
- Qian, B., Wang, L., Huang, D.X., Wang, W.L., Wang, X., 2009. An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Comput. Oper. Res.* 36 (1), 209–233.
- Qian, B., Wang, L., Huang, D.X., Wang, X., 2008. Scheduling multi-objective job shops using a memetic algorithm based on differential evolution. *Int. J. Adv. Manuf. Technol.* 35 (9–10), 1014–1027.
- Quintanilla, S., Pérez, Á., Lino, P., Valls, V., 2012. Time and work generalised precedence relationships in project scheduling with pre-emption: an application to the management of service centres. *Eur. J. Oper. Res.* 219 (1), 59–72.

- Shukla, S.K., Son, Y.J., Tiwari, M.K., 2008. Fuzzy-based adaptive sample-sort simulated annealing for resource-constrained project scheduling. *Int. J. Adv. Manuf. Technol.* 36 (9–10), 982–995.
- Su, W., Chow, M.Y., 2012. Performance evaluation of an EDA-based large-scale plug-in hybrid electric vehicle charging algorithm. *IEEE Trans. Smart Grid* 3 (1), 308–315.
- Sundarakani, B., De Souza, R., Goh, M., Wagner, S.M., Manikandan, S., 2010. Modeling carbon footprints across the supply chain. *Int. J. Prod. Res.* 128 (1), 43–50.
- Tang, L., Luh, P.B., Liu, J., Fang, L., 2002. Steel-making process scheduling using Lagrangian relaxation. *Int. J. Prod. Res.* 40 (1), 55–70.
- Voß, S., Witt, A., 2007. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: a real-world application. *Int. J. Prod. Res.* 105 (2), 445–458.
- Wang, L., Fang, C., 2012. An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem. *Comput. Oper. Res.* 39 (2), 449–460.
- Wang, L., Fang, C., Mu, C.D., Liu, M., 2013a. A Pareto-archived estimation-of-distribution algorithm for multiobjective resource-constrained project scheduling problem. *IEEE Trans. Eng. Manag.* 60 (3), 617–626.
- Wang, L., Wang, S., Liu, M., 2013b. A Pareto-based estimation of distribution algorithm for the multi-objective flexible job-shop scheduling problem. *Int. J. Prod. Res.* 51 (12), 3574–3592.
- Wang, L., Fang, C., Suganthan, P.N., Liu, M., 2014. Solving system-level synthesis problem by a multi-objective estimation of distribution algorithm. *Expert Syst. Appl.* 41 (5), 2496–2513.
- Wang, L., Wang, S., Xu, Y., 2012a. An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem. *Expert Syst. Appl.* 39 (5), 5593–5599.
- Wang, L., Zhou, G., Xu, Y., Liu, M., 2012b. An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. *Int. J. Adv. Manuf. Technol.* 60 (9–12), 1111–1123.
- Wang, S., Wang, L., Liu, M., Xu, Y., 2013a. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *Int. J. Prod. Econ.* 145 (1), 387–396.
- Wang, S., Wang, L., Xu, Y., Liu, M., 2013b. An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Int. J. Prod. Res.* 51 (12), 3778–3793.
- Yu, S., Wei, Y.M., Guo, H., Ding, L., 2014. Carbon emission coefficient measurement of the coal-to-power energy chain in China. *Appl. Energy* 114, 290–300.
- Zhang, B., Wang, Z., 2014. Inter-firm collaborations on carbon emission reduction within industrial chains in China: practices, drivers and effects on firms' performances. *Energy Econ.* 42, 115–131.
- Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* 3 (4), 257–271.