A Pareto-Archived Estimation-of-Distribution Algorithm for Multiobjective Resource-Constrained Project Scheduling Problem

Ling Wang, Chen Fang, Chun-Di Mu, and Min Liu

***Abstract*—In this paper, a Pareto-archived estimation-of-distribution algorithm (PAEDA) is presented for the multiobjective resource-constrained project scheduling problem with makespan and resource investment criteria. First, by combining the activity list and the resource list, an encoding scheme named activity-resource list is presented. Second, a novel hybrid probability model is designed to predict the most promising activity permutation and resource capacities. Third, a new sampling and updating mechanism for the probability model is developed to track the area with promising solutions. In addition, a Pareto archive is used to store the nondominated solutions that have been explored, and another archive is used to store the solutions for updating the probability model. The evolution process of the PAEDA is visualized showing the most promising area of the search space is tracked. Extensive numerical testing results then demonstrate that the PAEDA outperforms the existing methods.**

***Index Terms*—Estimation-of-distribution algorithm, makespan, multiobjective resource-constrained project scheduling problem (RCPSP), Pareto-archived, probability model, resource investment.**

I. Introduction

To maintain enterprise competitiveness, scheduling has become an important issue in project management. A manager needs to decide when to start each activity and how to allocate the resources. The resource-constrained project scheduling problem (RCPSP) is one of the most important scheduling problems in engineering management, where scarce resources are allocated to dependent activities over time for single-item or small batch production [1]. Many practical problems can be formulated as the RCPSP, such as shop scheduling problems [2]–[4] in manufacturing, system-level synthesis problems [5] in industrial electronics, and software project management [6] in software engineering. In the RCPSP, a project consists of a number of activities to be scheduled nonpreemptively. There are precedence relationships among activities specifying that certain activities cannot be started earlier than its predecessors. There are also renewable resources with constant capacities for each period of the project. Given the capacity of each renewable resource, the RCPSP is to find a reasonable utilization of the resources and an efficient schedule of the project activities to minimize the project makespan. The RCPSP has been shown to be NP-hard [7]. As a result, only small-sized problems can be solved optimally in an acceptable amount of time by conventional exact algorithms such as the branch-and-bound. For large-sized problems, heuristics are usually used to obtain suboptimal solutions within the given amounts of time and memory space. Comprehensive reviews on the RCPSP can be found in [1], [8]–[12].

With the emergencies of new technologies and new products, the complexity of project management has increased greatly during the past few decades. In practice, multiple objectives (such as makespan and resource investment) should be considered in project scheduling, and project scheduling can be regarded as a multiobjective optimization problem. Relevant results in this area include the following. Pulat and Horn [13] developed an enumerative and interactive algorithm for the time-resource tradeoff problem with two resources. Hapke *et al.* [14] presented a two-stage procedure to solve the multiobjective project scheduling problems with the criteria of makespan, resource utilization smoothness, maximum lateness, mean flow time, net present value, and project cost. In the first stage, the Pareto simulated annealing (PSA) was used to generate a set of approximately nondominated solutions. In the second stage, the set of the solutions generated was interactively analyzed by the light beam search procedure. Viana and Sousa [15] developed multiobjective versions of simulated annealing and tabu search to minimize makespan, weighted lateness of activities, and the violation of resource constraints. Al-Fawzan and Haouari [16] introduced the concept of schedule robustness, and developed a biobjective resource-constrained project scheduling model and a tabu search algorithm, where both makespan and robustness were taken into account. Abbasi *et al.* [17] also considered a biobjective RCPSP with the criteria of robustness and makespan, and developed a simulated annealing to minimize a weighted sum of the objective functions. Pollack-Johnson and Liberatore [18] incorporated quality considerations into the time-cost tradeoff analysis and presented a mixed integer linear programming model for the time-cost-quality tradeoff problem. Mokhtari *et al.* [19] developed an ant colony system for the

Manuscript received December 22, 2011; revised May 10, 2012, July 13, 2012, October 10, 2012, and November 28, 2012; accepted December 7, 2012. Date of publication January 22, 2013; date of current version July 13, 2013. This work was supported by the National Key Basic Research and Development Program of China under Program 2013CB329503, by the National Science Foundation of China under Grant 61174189, Grant 61025018, and Grant 60834004, by the Doctoral Program Foundation of Institutions of Higher Education of China under Grant 20100002110014, and by the National Science and Technology Major Project of China under Grant 2011ZX02504-008. Review of this manuscript was arranged by Department Editor S. (Sri) Talluri.

L. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wangling@mail.tsinghua.edu.cn).

C. Fang is with the Department of Automation, Tsinghua University, Beijing 100084, China and Industrial and Commercial Bank of China (e-mail: fangchen06@mails.tsinghua.edu.cn).

C.-D. Mu and M. Liu are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: muchd@mail.tsinghua.edu.cn; lium@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2012.2234463

discrete time-cost tradeoff problem. Rabbani *et al*. [20] presented a multiobjective particle swarm optimization algorithm for the project selection problem. Recently, Ballestín and Blanco [21] provided some theoretical and practical fundamentals for the multiobjective RCPSPs. They extended the PSA, the nondominated sorting genetic algorithm 2 (NSGA2) and the strength Pareto evolutionary algorithm 2 (SPEA2) to solve the RCPSP with makespan and resource investment criteria.

The estimation-of-distribution algorithm (EDA) is a newly developed method in stochastic optimization [22]. Different from the genetic algorithm which explicitly applies crossover and mutation operators to produce new individuals, EDA produces new individuals implicitly based on a probability model. It utilizes the statistical information to build a probability model to track the promising area. Then, it generates new individuals by sampling the probability model, and good individuals generated are selected to update the probability model. EDA has been applied to solve various problems, such as feature selection [23], flow-shop scheduling [24], nurse rostering [25], quadratic assignment problem (QAP) [26], multispeed planetary transmission design [27], inexact graph matching [28], software testing [29], RCPSP [30], and multimode RCPSP [31]. However, to the best of the authors' knowledge, EDA has not been adopted to solve the multiobjective resource-constrained project scheduling problem (MORCPSP) with makespan and resource investment criteria (MORCPSP–MS–RI) yet. In this paper, we shall present a Pareto-archived EDA (PAEDA) to solve the MORCPSP–MS–RI effectively. First, an encoding scheme named activity-resource list (ARL) is presented. Second, a novel hybrid probability model is designed to predict the most promising activity permutation and resource capacities. Third, a new sampling and updating mechanism for the probability model is developed to guide the EDA-based evolutionary search to track the most promising area. In addition, a Pareto archive (PA) is used to store the nondominated solutions that have been explored, and another archive is used to store the solutions for updating the probability model. The evolution process of the PAEDA is visualized showing the most promising area is tracked. And numerical testing results show that the PAEDA outperforms the existing methods. The PAEDA enriches the tool-kit for project scheduling, and offers the project manager a set of nondominated solutions to obtain a proper tradeoff between makespan and resource investment. The PAEDA may also be applied to solve other kinds of project scheduling problems and the related practical engineering management problems.

The rest of this paper is organized as follows. In Section II, the MORCPSP–MS–RI is mathematically formulated. In Section III, a brief introduction to the EDA is provided. In Section IV, the detailed PAEDA is presented to solve the MORCPSP–MS–RI followed by the numerical testing results provided in Section V. In Section VI, the practical implications of the PAEDA are discussed. Finally, some conclusions and future work are presented in Section VII.

II. Problem Formulation

The RCPSP is involved with scheduling a project that consists of J activities $\{1,\ldots,J\}$. Without loss of generality, two dummy activities 0 and $J+1$ are introduced to represent the start and the end of the project, respectively. Then, the set of all activities is denoted as $J^+ = \{0,\ldots,J{+}1\}$. For each activity $j \in J^+$, let d_j be its processing time and P_j be the set of its immediate predecessors. Thus, activity j cannot be started until all the activities in P_j are completed. There are K types of renewable resources. Once it is started, activity j occupies r_{jk} units of resource k in each period of its nonpreemptable duration. Note that the dummy activities have zero processing time (i.e., $d_0 = d_{J+1} = 0$) and do not use any resource (i.e., $r_0 k = r_{J+1}, k = 0$ for all k). The total amount of resource k used at time t, denoted as $r_k(t)$, is bounded a constant R_k (called the maximal per-period-availability). According to the maximal value of $r_k(t)$ within the production, we need to pay an investment cost c_k for each unit of that. Since there are two objective functions, namely makespan (MS) and resource investment (RI), the MORCPSP–MS–RI can be formulated as follows:

$$\min_{x_{jt}\in\{0,1\},j\in J^+,t=1,\ldots,T} [\mathrm{MS}, \mathrm{RI}] \tag{1}$$

$$\text{Subject to MS} = \sum_{t=\mathrm{EFT}_{J+1}}^{\mathrm{LFT}_{J+1}} t \cdot x_{J+1,t} \tag{2}$$

$$\mathrm{RI} = \sum_{k\in K} c_k \max_{t\in[0,T]} \{r_k(t)\} \tag{3}$$

$$r_k(t) = \sum_{j=1}^{J} \sum_{b=\max\{t,\mathrm{EFT}_j\}}^{\min\{t+d_j-1,\mathrm{LFT}_j\}} r_{jk} x_{jb} \tag{4}$$

$$\sum_{t=\mathrm{EFT}_j}^{\mathrm{LFT_j}} x_{jt} = 1, \;\; j \in J^+ \tag{5}$$

$$\sum_{t=\mathrm{EFT}_h}^{\mathrm{LFT_h}} t \cdot x_{ht} \le \sum_{t=\mathrm{EFT}_j}^{\mathrm{LFT_j}} (t-d_j) x_{jt}, \;\; j \in J^+; \;\; h \in P_j \tag{6}$$

$$\max_{j\in\{1,\ldots,J\}} \{r_{jk}\} \le r_k(t) \le R_k, \;\; k{=}1,2,\ldots,K; t=1,2,\ldots,T \tag{7}$$

$$x_{jt} = \begin{cases} 1, & \text{if activity } j \text{ finishes at time } t. \\ 0, & \text{otherwise.} \end{cases}, \quad j \in J^+ \tag{8}$$

where $x_{jt} \in \{0,1\}$ is a decision variable that describes whether activity j finishes at time t, as shown in (8); T is an upper bound for the makespan, i.e., $T = \sum_{j\in J^+} d_j$; LFT_j and EFT_j denote the latest finish time and the earliest finish time of activity j, respectively, which can be computed under the assumption that there is no resource constraint. Equation (1) implies that a biobjective minimization problem is addressed. Equations (2) and (3) define the makespan and the resource investment, respectively. Equation (4) describes the amount of resource k used at time t. Equation (5) guarantees that each activity will be processed exactly once. Equations (6) and (7) describe the precedence constraints among the activities and the renewable constraints on the resources, respectively.

Let $x = \{x_{jt}, j \in J^+, t = 1, \ldots, T\}$ be a solution and χ be the set of all solutions. For $x_1, x_2 \in \chi$ of such a biobjective

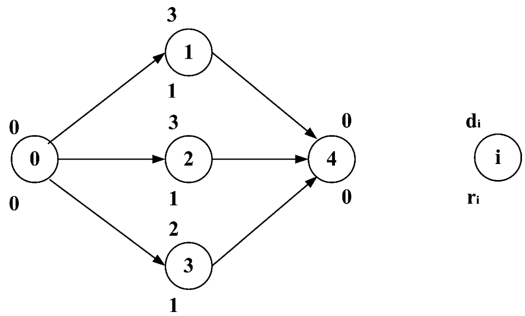

Fig. 1. Example of the MORCPSP-MS–RI.

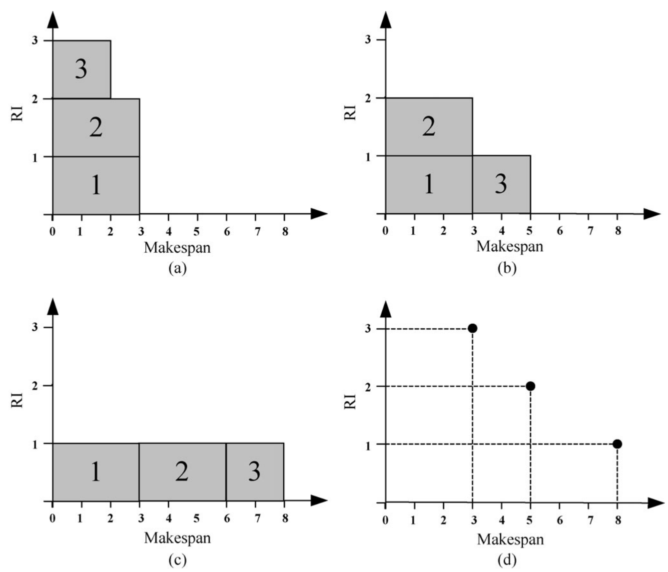

Fig. 2. Pareto optimal solutions for the example in Fig. 1.

optimization problem, we say x_1 dominates x_2 if and only if (iff)

$$\text{MS}(x_1) \leq \text{MS}(x_2) \ \text{ and } \ \text{RI}(x_1) \leq \text{RI}(x_2) \tag{9}$$

with at least one of the aforementioned inequalities being strict. It is denoted as $x_2 \prec x_1$. A solution x is said to be Pareto optimal iff there does not exist another solution $x' \in \chi$ s.t. $x \prec x'$. The set of all Pareto optimal solutions constitutes the Pareto set, which forms a front in the objective space (called the Pareto front). The goal of the MORCPSP–MS–RI is to find the Pareto set or part of the set.

An example of the MORCPSP–MS–RI with five activities is shown in Fig. 1, where activity 0 and activity 4 are two dummy activities. The project is described by an activity-on-node (AoN) network, in which a node represents an activity and an arc represents a precedence constraint between two activities. There is a resource with $R_1 = 3$ and $c_1 = 1$. Three Pareto optimal solutions are shown in Fig. 2(a)–(c). Their makespan values and resource investments are then given in Fig. 2(d).

III. Estimation-of-Distribution Algorithm

The estimation-of-distribution algorithm (EDA) provides a framework for a group of probability distribution-based algorithms. The idea is to predict and track the area with promising

Step 1: Construct an initial probability model.

Step 2: Sample a population via the probability model.

Step 3: (Optional) Perform the local search.

Step 4: Construct the elite set from the population and update the probability model.

Step 5: Go to Step 2 unless a certain stopping criterion is met.

Fig. 3. Procedure of EDA.

solutions in the search space by using a probability model. The procedure of EDA is shown in Fig. 3. Note that different probability models should be adopted for different problems. In the next section, a special PAEDA will be presented for solving the MORCPSP–MS–RI effectively.

IV. Pareto-Archived EDA

A. Solution Encoding and Initial Population

A solution is encoded as an activity-resource list (ARL), i.e., ARL = [λ, π], where λ = ($\lambda_1, \ldots, \lambda_J$) is the activity list (AL) representing the sequence of activities, and $\pi = (\pi_1, \ldots, \pi_K)$ is the resource list (RL) denoting the capacities of resources. Note that Kolisch and Hartmann [9] used experimental results to show that the AL representation can produce good solutions to the RCPSP. Because the MORCPSP–MS–RI can be regarded as a generalization of the RCPSP, the aforementioned ARL representation is also expected to produce good solutions.

Once an ARL is given, the serial schedule generating scheme (SSGS) [32] is employed to construct a schedule effectively. To be specific, the activities are scheduled in the AL order at their earliest possible start times. Note that the resource constraints and the precedence constraints should also be satisfied. More details of the SSGS were provided in [31].

The initial population is generated as follows. First, an available amount of each resource is uniformly randomly chosen to generate an RL. Then, it follows an iterative procedure to generate the AL. Let S^l be the set of the selected activities after the lth iteration. Then, $S^l = \{0\}$ if $l = 1$, which means that only the dummy activity 0 is selected at the beginning. Let $P(j)$ be the set of predecessors of activity j in the AoN. Then, $S_E^l = \{j \in J^+/S^l, P(j) \subseteq S^l\}$ is used to define the set of activities which have not been selected yet but can be started after some of the selected activities are finished. The set S_E is called the eligible set in the lth iteration. In each iteration, an activity $j \in S_E^l$ is added to S^l with probability η_j, where [33]

$$\eta_j = (\mu_j + 1) \Big/ \sum_{i \in S_E} (\mu_i + 1) \tag{10}$$

$$\mu_j = \max_{i \in S_E} \text{LFT}_i - \text{LFT}_j. \tag{11}$$

It is also known as the latest-finish-time (LFT) priority rule [34]. Then, $l = l+1$, and it enters the next iteration. The aforementioned procedure will be repeated until all the activities are selected. Thus, an AL will be generated. The method is also called the regret-based biased random sample method (RBBRS) [35], which will be utilized to generate a few solutions to constitute the initial population Init_POP.

B. Hybrid Probability Model

As discussed in [36], the choice of the probability model will affect the performance of EDA greatly. In this regard, a hybrid probability model is used, which contains two parts, namely a probability matrix and a set of probability vectors. More details are provided as follows.

1) (1) Probability matrix Define:

$$M\_\text{act}(g) = \begin{pmatrix} \alpha_{11}(g) & \cdots & \alpha_{1J}(g) \\ \vdots & \ddots & \vdots \\ \alpha_{J1}(g) & \cdots & \alpha_{JJ}(g) \end{pmatrix} \quad (12)$$

where $\alpha_{ji}(g)$ is the probability to place activity j at the ith position in the AL in the gth generation. The initial probability matrix is

$$M\_\text{act}(0) = \begin{pmatrix} 1/J & \cdots & 1/J \\ \vdots & \ddots & \vdots \\ 1/J & \cdots & 1/J \end{pmatrix} \quad (13)$$

which implies that all the activities can be selected uniformly randomly into the AL.

2) (2) Probability Vector Define:

$$V\_\text{res}_k(g) = [\beta_{L_k}, \beta_{L_{k+1}}, \ldots, \beta_{U_k}], \quad k = 1, 2, \ldots, K \quad (14)$$

where L_k and U_k are the lower and upper bounds of the capacity of resource k, respectively; $\beta_{m(k)}$, which is also denoted as V_res_{km}, is the probability that the capacity of resource k is $m(k)$, where $L_k \leq m(k) \leq U_k$. In the PAEDA, we use $L_k = \max_{j \in \{1,\ldots,J\}}\{r_{jk}\}$ (the minimal amount of resource k in a feasible solution) and $U_k = R_k$ (the maximal amount of resource k in a project). The probability vectors are initialized as follows to ensure that the whole solution space is sampled uniformly.

$$V\_\text{res}_k(0) = \left[\frac{1}{U_k - L_k + 1}, \frac{1}{U_k - L_k + 1}, \ldots, \frac{1}{U_k - L_k + 1}\right] \quad k = 1, 2, \ldots, K. \quad (15)$$

C. Probability-Model-Based Sampling Mechanism

In each generation of the PAEDA, the individuals in the population are obtained by sampling the ALs and RLs using the probability models. To be specific, the AL is first sampled using the probability matrix $M\_\text{act}(g)$ over the set of eligible activities S_E. The probability for selecting activity j at position i of the AL, Pa_{ji}, is

$$Pa_{ji} = \alpha_{ji} \Big/ \sum_{h=1}^{S_E} \alpha_{hi}. \quad (16)$$

If activity j has already been placed at a certain position, the whole jth row $(\alpha_{J1}, \alpha_{J2}, \ldots, \alpha_{jJ})$ of the probability matrix M_act is set to 0. As a result, each activity is selected

```
Procedure Pareto Archive Updating
  UA=Φ;
  For (i=1,2,…,|POP|)
      flag=1;
      For (j=1,2,…,|PA|)
          If (I_i ≺ A_j  or  I_i = A_j)
              flag=0;
              break;
      If (flag==1)
          For (j=1,2,…,|PA|)
              If (A_j ≺ I_i)
                  Remove A_i from PA;
          Add I_i into PA and UA;
```

Fig. 4. Procedure of archive updating.

only once in an AL. When all the activities are selected, a sample of AL is obtained. Then, the RL is sampled using V_res_k, $k = 1, \ldots, K$. For example, suppose that $K = 3$, $V_{\text{res}_1} = [0.1, 0.3, 0.6]$, $V_{\text{res}_2} = [0.2, 0.4, 0.1, 0.1, 0.2]$, $V_{\text{res}_3} = [0.2, 0.4, 0.3, 0.1]$, $L_1 = 1$, $L_2 = 1$, $L_3 = 2$, $U_1 = 3$, $U_2 = 5$, and $U_3 = 5$. According to the selecting probability, it may get $m(1) = 3$, $m(2) = 1$, and $m(3) = 4$. Thus, RL is $\pi = [3, 1, 4]$.

D. PA and Updating Archive

In the PAEDA, a PA is used to store all the nondominated solutions that have been explored. At the beginning, the PA is constructed with the initial population. In each subsequent generation, every solution I_j $(j = 1, 2, \ldots, |\text{POP}|)$ in the new population POP will be compared with each member A_j $(j = 1, 2, \ldots, |\text{PA}|)$ in the PA. The new solutions that are not dominated by the current PA will be added into the PA, while the solutions in the current PA which are dominated by the newly added ones will be discarded. Moreover, all those nondominated solutions that have been newly found will be stored in an updating archive (UA), which will be used to update the probability model. The aforementioned procedure is summarized in Fig. 4.

E. Updating Mechanism for the Probability Model

In each generation, the hybrid probability model is updated based on the UA by the following strategy:

$$M\_\text{act}_{ji}(g+1) = (1-\theta_1) \cdot M\_\text{act}_{ji}(g) + \frac{\theta_1}{|\text{UA}|} \sum_{n=1}^{|\text{UA}|} L_{ji}^n, (1 \leq i, j \leq J) \quad (17)$$

$$V\_\text{res}_{km}(g+1) = (1-\theta_2) \cdot V\_\text{res}_{km}(g) + \frac{\theta_2}{|\text{UA}|} \sum_{n=1}^{|\text{UA}|} R_{km}^n, (1 \leq k \leq K, L_k \leq m \leq U_k) \quad (18)$$

where θ_1 and θ_2 denote the learning rates of $M\_\text{act}(g)$ and $V\_\text{res}_k(g)$, $k = 1, \ldots, K$, respectively. L_{ji}^n and R_{km}^n are the indicator functions. $L_{ji}^n = 1(0)$, if activity j is placed at position i in the nth individual of UA (otherwise). $R_{km}^n = 1(0)$, if the amount of resource k used to process the project is m

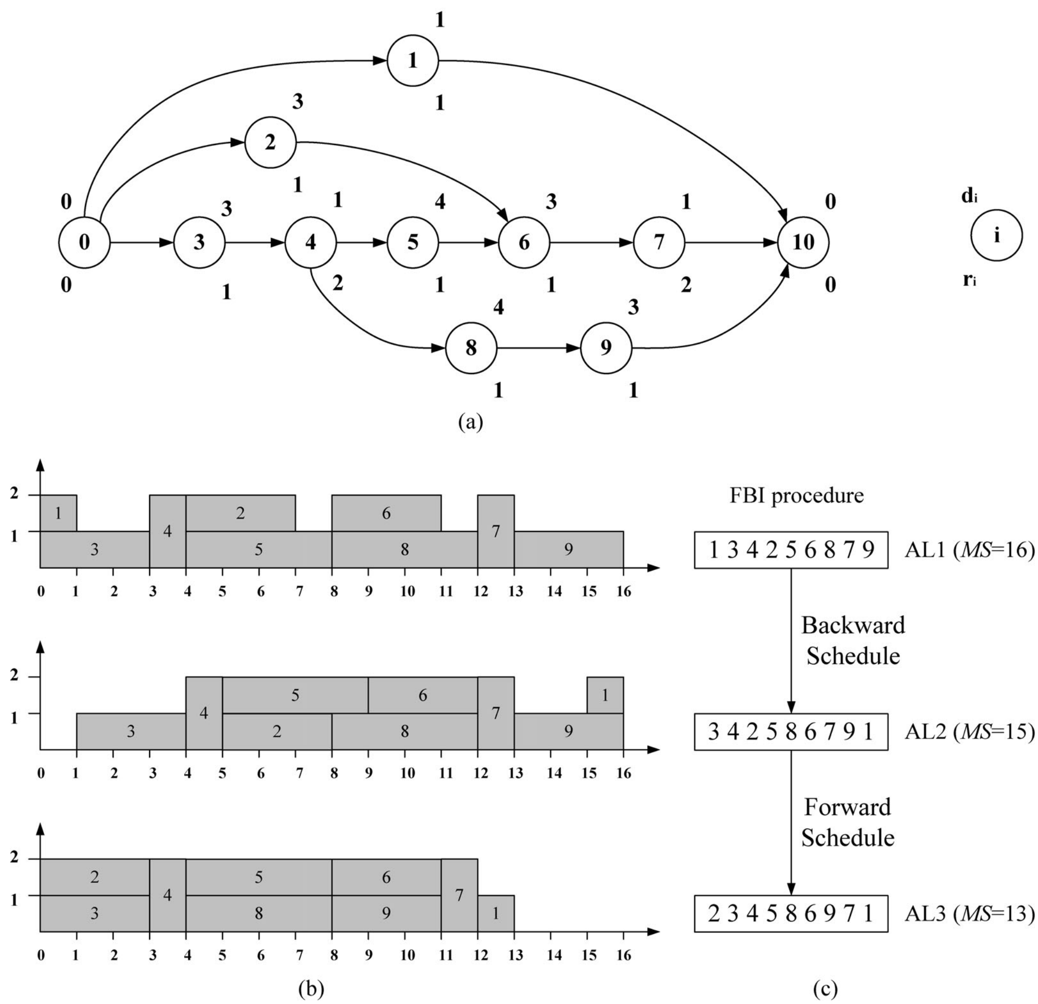

Fig. 5. Local search strategy.

(otherwise). Note that if the UA is empty, the hybrid probability model will not be updated.

F. Local Search Strategy

In the PAEDA, the well-known forward–backward improvement (FBI) [37] is adopted as a local search strategy to improve the individuals. Generally, the FBI makes use of the forward and backward SSGS iteratively to improve makespan. An example of the local search is given in Fig. 5. The AoN network of the project is shown in Fig. 5(a). The schedules generated in the FBI procedure are provided in Fig. 5(b). First, the activities are scheduled backward in a descending order of the completion times. Then, the activities are scheduled forward in an ascending order of the start times. The procedure is repeated until no further makespan improvement can be achieved. The corresponding ALs are given in Fig. 5(c), where the makespan is reduced from 16 to 13.

G. Procedure of the PAEDA

The procedure of the PAEDA for the MORCPSP–MS–RI is summarized as follows. First, the probability matrix $M\_\text{act}(g)$ and vectors $V\_\text{res}_k(g)$ $(k = 1,2, \ldots, K)$ are initialized using the initial population Init_POP at $g = 0$. Then, the new population POP(g) is generated by the probability-model-based sampling mechanism. After all the individuals are evaluated by the SSGS, the FBI is used to improve the individuals, and POP(g) is updated accordingly. Then, PA and UA are updated based on POP(g). The probability matrix $M\_\text{act}(g)$ and vectors $V\_\text{res}_k(g)$ ($k = 1,2, \ldots ,K$) are updated using UA. The aforementioned procedure is repeated with $g = g{+}1$ until a stopping criterion is met. More intuitive illustration of the procedure of the PAEDA is shown in Fig. 6.

V. Numerical Experiments

A. Experimental Setting

The PAEDA is implemented in Visual C++ 2005. All the experiments in this section are performed on an IBM Thinkpad X60 with a Core T5600/1.83 GHz processor. The subsets j30 and j120 from the standard benchmark PSPLIB [38] are used for testing. The subset j30 consists of 480 projects with 30 nondummy activities, and j120 consists of 600 projects with 120 nondummy activities. The PSPLIB is designed by a full-factor design method with different network complexities (NC), resource factors (RF), and resource strengths (RS). It covers the

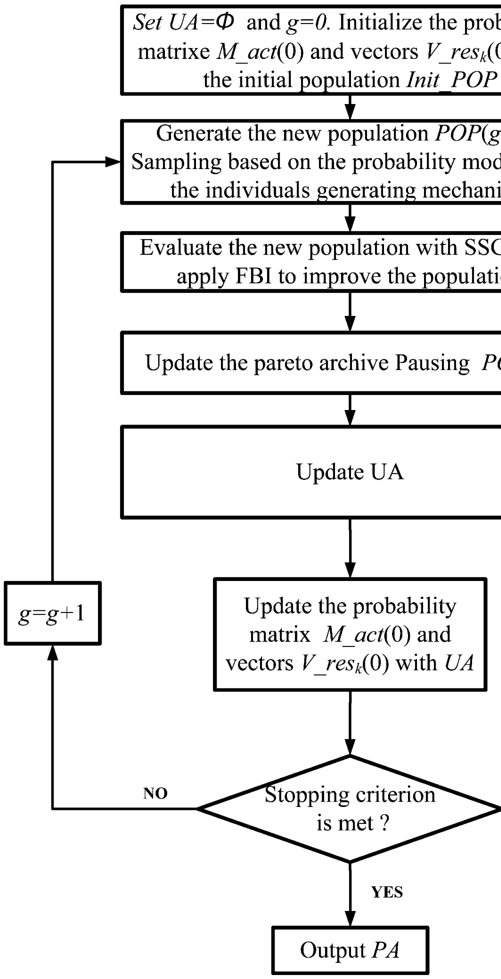

Fig. 6. Procedure of the PAEDA.

combinations of project parameters, i.e., NC, RF, and RS arising in the real-world project scheduling problems. The PSPLIB is widely used to test and to compare different project scheduling methods. It can also be extended to other engineering management and scheduling problems including the well-known manufacturing scheduling problems. However, the PSPLIB is designed only for the makespan criterion and there are no specific cost values for the resource investment criterion. Without loss of generality, we set all the cost values as 1 in the MORCPSP–MS–RI. In the experiments, we set $|\mathrm{POP}| = 100$, $\theta_1 = 0.1$, and $\theta_2 = 0.05$, and stop the PAEDA when 1000, 5000, 50 000 schedules are generated in total.

B. Performance Metrics

The following two performance metrics are employed to compare the PAEDA with other algorithms.

1) The size of the approximate Pareto set.
The approximate Pareto set is composed of all the non-dominated solutions that have been explored by an algorithm. This set will be delivered to the project manager for final decision making. Let M be the size of the set. Note that a sound algorithm normally has a large M, which means that the algorithm can provide more choices to the manager.

2) The coverage of the approximate Pareto sets obtained by two algorithms [39].
Suppose there are two approximate Pareto sets, i.e., $X', X'' \subseteq \chi$, which are obtained by applying two different algorithms. The coverage of X' on X'' is defined as follows:

$$C\left(X', X''\right) = \frac{\left|\left\{a'' | a'' \in X'', \exists a' \in X' : a' \prec a'' \mathrm{or} a' = a''\right\}\right|}{|X''|}. \tag{19}$$

If $C(X', X'') = 1$, it means that all the solutions in X'' are dominated by or equal to the solutions in X'. If $C(X', X'') = 0$, it indicates that none of the solutions in X'' is dominated by or equal to the solutions in X'. Note that both $C(X', X'')$ and $C(X'', X')$ need to be considered to carry out a comparison, since the sum of $C(X', X'')$ and $C(X'', X')$ may not be equal to 1. If $C(X', X'')$ is larger than $C(X'', X')$, it can be concluded that X' is better than X'' in sense of the Pareto dominance. That is, X' is more helpful for the manager to make a better decision. As a result, the algorithm producing X' is superior to that producing X''.

C. Evolution Process of the Probability Model

In this section, the instance j3014_7.RCP is used to visualize the evolution process of the probability model in the PAEDA.

In Fig. 7, the evolution process of the probability matrix $M\_\mathrm{act}(t)$ is illustrated. It can be seen that all the elements of the probability matrix are equal to each other at the beginning of the evolution. As the algorithm evolves, the quantities of most elements become smaller, while the quantities of some other elements become larger. It means that the probability of placing a certain activity at a certain position of the AL becomes larger. Similar phenomena can be observed in the probability vectors as shown in Fig. 8. The quantities of some elements become larger during the evolution process of the probability vectors, which means that the probability turns larger to set the capacity of each resource to a certain related value. Consequently, the search space of the PAEDA is reduced. Moreover, the distribution information of the activities and resources can help the manager choose the suitable activity locations and resource capacities. For example, it can be seen from Fig. 8 that the most promising capacities of resource 1, 2, 3, and 4 are 13, 12, 10, and 11, respectively.

In Fig. 9, the Pareto fronts of the instance j3014_7.RCP with different numbers of schedules generated are shown. It can be seen that the Pareto front is improved as more schedules are generated, which indicates that the PAEDA can keep on finding better solutions. Moreover, the size of the approximate Pareto set is 22, 28, and 31 when 1000, 5000, 50 000 schedules are generated, respectively. This implies that the PAEDA can provide more approximate Pareto optimal solutions when more computing budget is spent. Furthermore, it can be seen that the area with promising solutions is well tracked by the hybrid probability model.

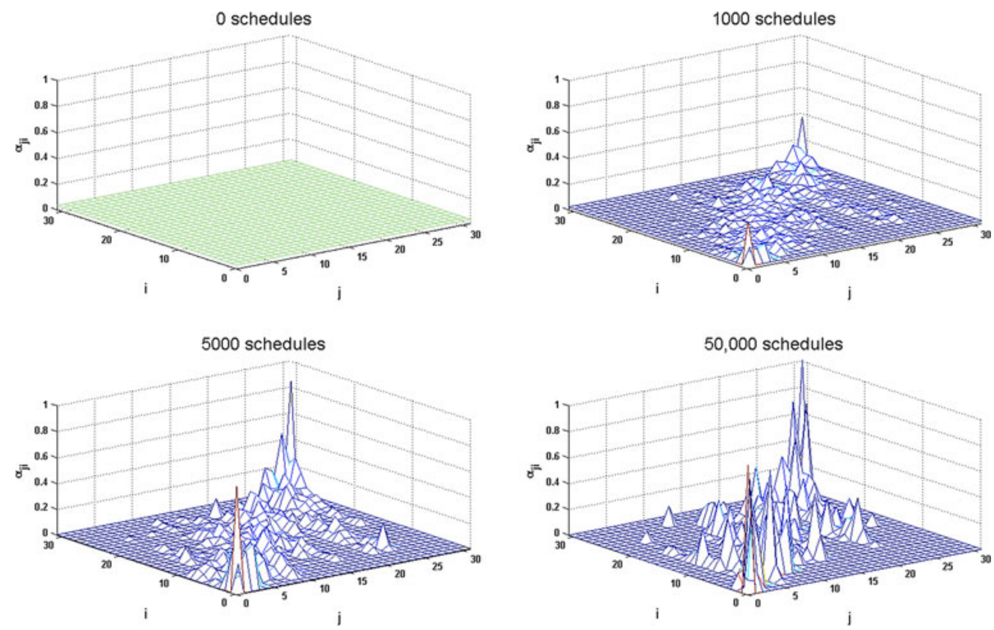

Fig. 7. Evolution process of the probability matrix.

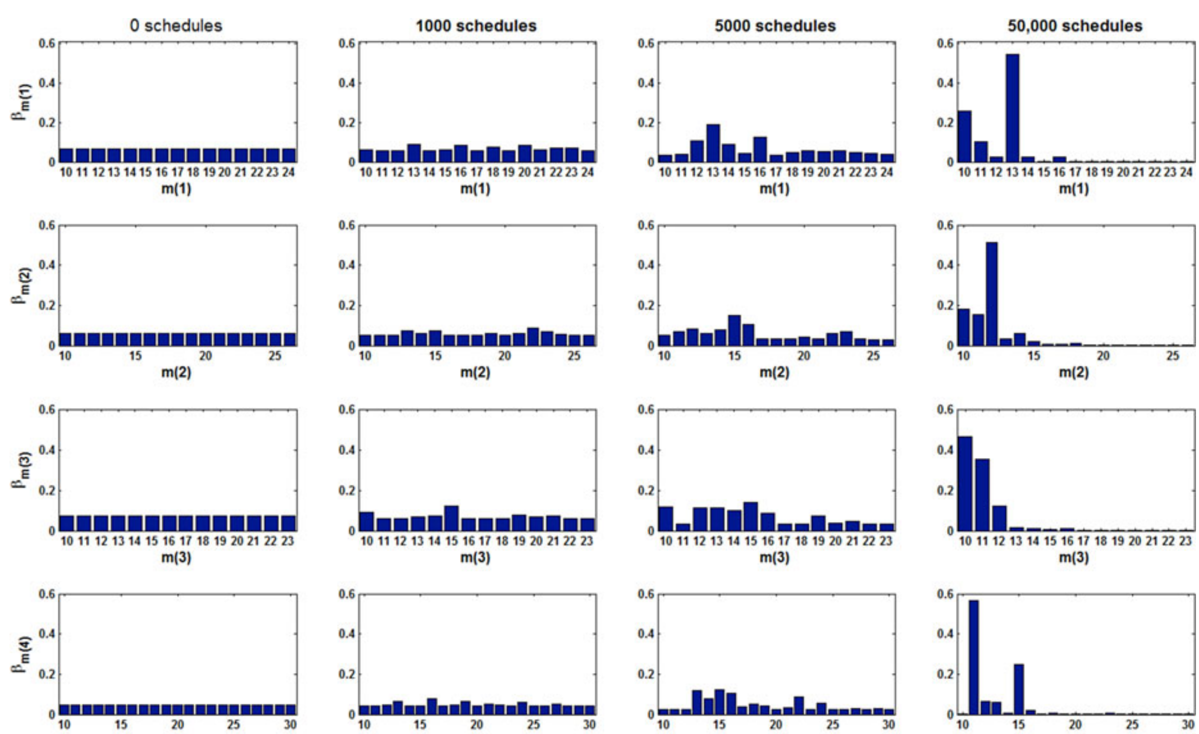

Fig. 8. Evolution process of the probability vectors.

D. Comparison of Different Algorithms

Next, the PAEDA is compared with the random search algorithm (RAND) and the NSGA2. The original NSGA2 was proposed by Deb *et al*. [40] for solving continuous optimization problems. Recently, Ballestín and Blanco [21] extended the NSGA2 to solve the MORCPSP–MS–RI. They established some theoretical and practical fundamentals for correct algorithmic developments, and concluded that the NSGA2 performs better than the PSA and SPEA2. So, we select the extended NSGA2 [21] for comparison.

The average sizes of the approximate Pareto sets for j30 and j120 are shown in Tables I and II, respectively. It can be seen that the PAEDA delivers larger approximate Pareto sets than the RAND in all the cases. The average size of the Pareto set of the PAEDA is smaller than that of the NSGA2 when 1000 or 5000 schedules are generated, but is larger than that of the NSGA2 when 50 000 schedules are generated. So, the PAEDA

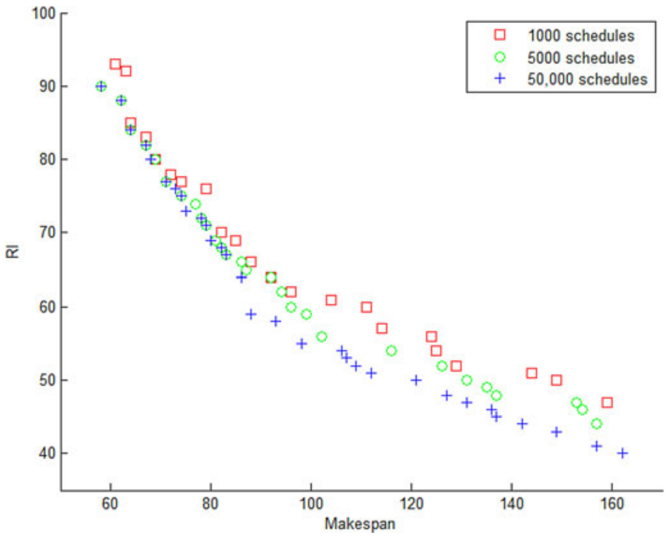

Fig. 9. Pareto front with different computing budget.

TABLE I
AVERAGE SIZE OF THE APPROXIMATE PARETO SET FOR j30

Algorithms	1000 schedules	5000 schedules	50,000 schedules
RAND	11.40	13.76	16.63
PAEDA	11.52	14.35	17.10
NSGA2	15.93	16.82	14.90

TABLE II
AVERAGE SIZE OF THE APPROXIMATE PARETO SET FOR j120

Algorithms	1000 schedules	5000 schedules	50,000 schedules
RAND	18.01	24.32	33.79
PAEDA	19.99	27.53	35.84
NSGA2	25.01	31.17	35.60

is better than the NSGA2 to keep the population diverse as the algorithm evolves. Besides, the PAEDA can obtain more nondominated solutions when a larger number of schedules are generated.

Then, the box plots are used to visualize the distribution of the C values for comparison. The RAND, PAEDA, and NSGA2 are compared in pairs using the C metric. For each pair of algorithms (A and B), we compute all the C values of 480 and 600 instances for j30 and j120, respectively. The box plots with different computing budget for j30 and j120 are shown in Figs. 10 and 11, respectively. In each subfigure, it contains three box plots which represent the distributions of the C values for pairs of algorithms when 1000, 5000, 50 000 schedules are generated, respectively. The upper and the lower ends of the box indicate the upper and the lower quartiles. The line within the box indicates the median. The ends of the whiskers represent the 5th percentile and the 95th percentile, respectively. The data that are not included between the whiskers are outliers which are plotted as dots.

From Figs. 10 and 11, it can be seen that the PAEDA outperforms the NSGA2, which in turn outperforms the RAND on both j30 and j120 for all the three cases. Taking j120 with

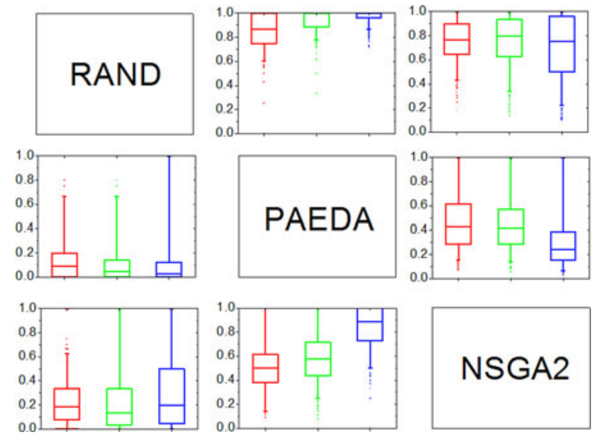

Fig. 10. Box plots of j30 based on the C metric.

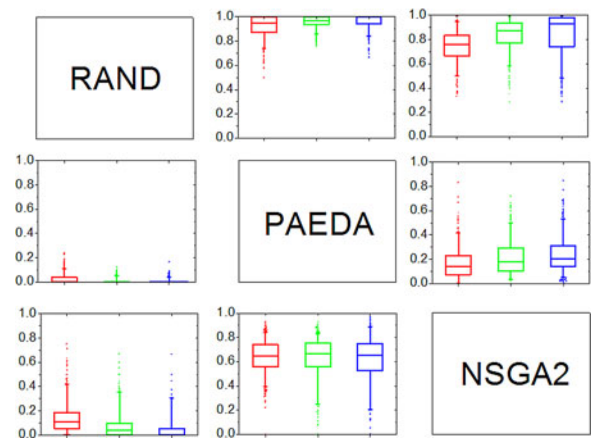

Fig. 11. Box plots of j120 based on the C metric.

50 000 schedules as an example, it can be seen from Fig. 11 that $C(\text{PAEDA, NSGA2}) = 0.62$ and $C(\text{NSGA2, PAEDA}) = 0.24$, which means that 62% of the nondominated solutions found by the NSGA2 are dominated by or equal to those found by the PAEDA, while only 24% of the nondominated solutions found by the PAEDA are dominated by or equal to those found by the NSGA2. So, the PAEDA is better than the NSGA2 to get better nondominated solutions in solving the MORCPSP–MS–RI. In practice, we would suggest to run both the PAEDA and the NSGA2, and then combine their results together to find an even larger set of nondominated solutions.

VI. MANAGERIAL IMPLICATIONS

First, this research work shows that the utilization of statistical information may identify the most promising activity sequence and the combination of resource capacities. With the hybrid probability model, it can provide the project manager insight into the distribution of the project activities and resources. Then, he/she can place activities at suitable positions in the AL and choose suitable capacities for the resources, which is helpful for better decision-making in engineering management.

Second, the PAEDA aims at minimizing both makespan and resource investment. The PA helps the project manager to identify the nondominated plans and to understand the relationship between makespan and resource investment. The PAEDA outputs a set of nondominated solutions. So, the project manager can choose a suitable solution to make a flexible decision according to his/her preference, e.g., the due date of the project, or the maximum resource investment. The multiobjective handling technique in the PAEDA may also be helpful for practical applications to the multiobjective optimization problems with three or more objectives.

Third, the performance of the PAEDA has been demonstrated by numerical comparison with the best existing algorithm NSGA2. Because the RCPSP can be generalized to various other scheduling problems, such as the job shop and flow shop scheduling [2]–[4], the system-level synthesis [5], and the software project management [6], the PAEDA can be extended to solve other multiobjective scheduling problems.

Moreover, many algorithms have already been developed for solving the RCPSP. But it lacks some general software for real engineering management practice. The PAEDA enriches the tool-kit of multiobjective project scheduling techniques. It is really important to develop some software based on the PAEDA for multiobjective project scheduling problems.

VII. Conclusion

In this paper, the multiobjective RCPSP with makespan and resource investment criteria is considered. To the authors' best knowledge, this is the first work to design an EDA-based approach for solving the MORCPSP–MS–RI effectively. An activity-resource list is used to encode the solution. A hybrid probability model is designed to predict the most promising area in the solution space. A sampling and updating mechanism is used for the model to track the promising search area. Besides, a PA is used to store the nondominated solutions that have been explored, and another archive is used to store the Pareto optimal solutions newly found for updating the probability model. The numerical testing results and comparisons are presented, which show that the PAEDA performs well in solving the MORCPSP–MS–RI. The further work is to study the impact of NC, RF, and RS on the problem solutions, and to develop more effective algorithms by using structure information. Moreover, it is interesting to investigate the possible extension of the PAEDA for solving scheduling problems in manufacturing and system-level synthesis problems in industrial electronics.

Acknowledgment

The authors would like to thank the Department Editor Prof. T. Srinivas and the anonymous reviewers for their valuable comments and suggestions, and are also grateful to Prof. P. B. Luh, Prof. Q. Jia, and Dr. W. Wang for their help in improving the linguistic quality.

References

[1] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *Eur. J. Oper. Res.*, vol. 112, no. 1, pp. 3–41, 1999.

[2] T. C. Wong, F. T. S. Chan, and L. Y. Chan, "A resource-constrained assembly job shop scheduling problem with lot streaming technique," *Comput. Ind. Eng.*, vol. 57, no. 3, pp. 983–995, 2009.

[3] S. Voß and A. Witt, "Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application," *Int. J. Prod. Econ.*, vol. 105, no. 2, pp. 445–458, 2007.

[4] W. O. Rom, O. I. Tukel, and J. R. Muscatello, "MRP in a job shop environment using a resource constrained project scheduling model," *Omega*, vol. 30, no. 4, pp. 275–286, 2002.

[5] A. Gerstlauer, C. Haubelt, A. D. Pimentel, T. P. Stefanov, D. D. Gajski, and J. Teich, "Electronic system-level synthesis methodologies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1517–1530, Oct. 2009.

[6] E. Alba and J. F. Chicano, "Software project management with GAs," *Inform. Sci.*, vol. 177, no. 11, pp. 2380–2401, 2007.

[7] J. Blazewicz, J. K. Lenstra, and K. Rinnooy, "Scheduling subject to resource constraints: Classification and complexity," *Discr. Appl. Math.*, vol. 5, no. 1, pp. 11–24, 1983.

[8] E. L. Demeulemeester and W. Herroelen, *Project Scheduling: A Research Handbook*. Kluwer: Academic Pub, 2002.

[9] R. Kolisch and S. Hartmann, " Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis," in *Project Scheduling, Recent Models, Algorithms and Applications*, J. Weglarz, Ed. Berlin: Kluwer, 1999, pp. 147–178.

[10] R. Kolisch and R. Padman, "An integrated survey of deterministic project scheduling," *Omega*, vol. 29, no. 3, pp. 249–272, 2001.

[11] K. Neumann, C. Schwindt, and J. Zimmermann, *Project Scheduling With Time Windows and Scarce Resources*, 2nd ed. Berlin: Springer, 2003.

[12] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 1–14, 2010.

[13] P. S. Pulat and S. J. Horn, "Time-resource tradeoff problem," *IEEE Trans. Eng. Manage.*, vol. 43, no. 4, pp. 411–417, Nov. 1996.

[14] M. Hapke, A. Jaszkiewicz, and R. Słowiński, "Interactive analysis of multiple-criteria project scheduling problems," *Eur. J. Oper. Res.*, vol. 107, no. 2, pp. 315–324, 1998.

[15] A. Viana and J. Pinho de Sousa, "Using metaheuristics in multiobjective resource constrained project scheduling," *Eur. J. Oper. Res.*, vol. 120, no. 2, pp. 359–374, 2000.

[16] M. A. Al-Fawzan and M. Haouari, "A bi-objective model for robust resource-constrained project scheduling," *Int. J. Prod. Econ.*, vol. 96, no. 2, pp. 175–187, 2005.

[17] B. Abbasi, S. Shadrokh, and J. Arkat, "Bi-objective resource-constrained project scheduling with robustness and makespan criteria," *Appl. Math. Comput.*, vol. 180, no. 1, pp. 146–152, 2006.

[18] B. Pollack-Johnson and M. J. Liberatore, "Incorporating quality considerations into project time/cost tradeoff analysis and decision making," *IEEE Trans. Eng. Manage.*, vol. 53, no. 4, pp. 534–542, Nov. 2006.

[19] H. Mokhtari, R. Baradaran Kazemzadeh, and A. Salmasnia, "Time-cost tradeoff analysis in project management: An ant system approach," *IEEE Trans. Eng. Manage.*, no. 99, pp. 1–8, Feb. 2011.

[20] M. Rabbani, M. A. Bajestani, and G. B. Khoshkhou, "A multi-objective particle swarm optimization for project selection problem," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 315–321, 2010.

[21] F. Ballestín and R. Blanco, "Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems," *Comput. Oper. Res.*, vol. 38, no. 1, pp. 51–62, 2011.

[22] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Netherlands: Springer, 2002.

[23] Y. Saeys, S. Degroeve, D. Aeyels, Y. Van de Peer, and P. Rouz, "Fast feature selection using a simple estimation of distribution algorithm: A case study on splice site prediction," *Bioinformatics*, vol. 19, no. 2, pp. 179–188, 2003.

[24] B. Jarboui, M. Eddaly, and P. Siarry, "An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems," *Comput. Oper. Res.*, vol. 36, no. 9, pp. 2638–2646, 2009.

[25] U. Aickelin, E. K. Burke, and J. Li, "An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering," *J. Oper. Res. Soc.*, vol. 58, no. 12, pp. 1574–1585, 2007.

[26] Q. Zhang, J. Sun, E. Tsang, and J. Ford, "Combination of guided local search and estimation of distribution algorithm for quadratic assignment problems," in *Proc. Bird Feather Workshop Genetic Evolutionary Comput. Conf.*, Chicago, IL, Jul. 12–16, 2003, pp. 42–48.

[27] P. A. Simionescu, D. Beale, and G. V. Dozier, "Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm," *J. Mech. Design*, vol. 128, pp. 108–115, 2006.

[28] R. M. Cesar. Jr., E. Bengoetxea, I. Bloch, and P. Larrañaga, "Inexact graph matching for model-based recognition: Evaluation and comparison

of optimization algorithms," *Pattern Recogn.*, vol. 38, no. 11, pp. 2099–2113, 2005.
[29] R. Sagarna and J. A. Lozano, "On the performance of estimation of distribution algorithms applied to software testing," *Appl. Artif. Intell.*, vol. 19, no. 5, pp. 457–489, 2005.
[30] L. Wang and C. Fang, "A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2451–2460, 2012.
[31] L. Wang and C. Fang, "An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem," *Comput. Oper. Res.*, vol. 39, no. 2, pp. 449–460, 2012.
[32] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation," *Eur. J. Oper. Res.*, vol. 90, no. 2, pp. 320–333, 1996.
[33] R. Kolisch, *Project Scheduling Under Resource Constraints: Efficient Heuristics for Several Problem Classes*. Heidelberg: Springer Verlag, 1995.
[34] E. Davis and J. Patterson, "A comparison of heuristic and optimum solutions in resource-constrained project scheduling," *Manage. Sci.*, vol. 21, no. 8, pp. 944–955, 1975.
[35] R. Kolisch and A. Drexl, "Adaptive search for solving hard project scheduling problems," *Nav. Res. Log.*, vol. 43, no. 1, pp. 23–40, 1996.
[36] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. New York: Springer-Verlag, 2006.
[37] K. Y. Li and R. J. Willis, "An iterative scheduling technique for resource-constrained project scheduling," *Eur. J. Oper. Res.*, vol. 56, no. 3, pp. 370–379, 1992.
[38] R. Kolisch and A. Sprecher, "PSPLIB—A project scheduling problem library," *Eur. J. Oper. Res.*, vol. 96, pp. 205–216, 1996.
[39] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

Ling Wang received the B.Sc. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999, respectively.

Since 1999, he has been with the Department of Automation, Tsinghua University, where he became a Full Professor in 2008. His current research interests include intelligent optimization and scheduling. He has authored five academic books and more than 200 refereed papers.

Dr. Wang won the Outstanding Paper Award at the International Conference on Machine Learning and Cybernetics (ICMLC'02) in 2002, the Best Paper Award at the International Conference on Intelligent Computing (ICIC'11) in 2011, and the Top Cited Article Award by *Engineering Applications of Artificial Intelligence* (Elsevier). He also was the recipient of the Science and Technology Award of Beijing City in 2008, and the National Natural Science Award (First Place in 2003 and Second Place in 2007) nominated by the Ministry of Education of China. He became the Rising Star of Science and Technology of Beijing City in 2004, the Academic Young Talent of Tsinghua University in 2009, and the Program for New Century Excellent Talents in University by the MOE of China in 2009. He is the Co-Editor-in-Chief for *The Open Operational Research Journal*, and the editorial board member for several journals, including *Memetic Computing*, *Swarm and Evolutionary Computation*, *International Journal of Automation and Control*, and *International Journal of Artificial Intelligence and Soft Computing*. He also has been the reviewer for many IEEE journals like the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the IEEE *Computational Intelligence Magazine*, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTION ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B AND PART C.

Chen Fang received the B.Sc. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2012.

He is currently with the Industrial and Commercial Bank of China. His main research interests include project scheduling, scheduling theory, and intelligent optimization method.

Chun-Di Mu received the B.S. degree in control theory and control engineering from Tsinghua University, Beijing, China, in 1970.

She is currently a Professor with the Department of Automation, Tsinghua University. Her main research interests include optimization, control, and applications.

Min Liu (M'11) received the Ph.D. degree from Tsinghua University, Beijing, China, in 1999.

He is currently a Professor with the Department of Automation, Tsinghua University, an Associate Director of Automation Science and Technology Research Department of Tsinghua National Laboratory for Information Science and Technology, the Director of Control and Optimization of Complex Industrial Process, Tsinghua University, the Director of China National Committee for Terms in Automation Science and Technology, and the Director of Intelligent Optimization Committee of China Artificial Intelligence Association. His main research interests include optimization scheduling of complex manufacturing process and intelligent operational optimization of complex manufacturing process or equipment. He led more than 20 important research projects including the project of the National 973 Program of China, the project of the National Science and Technology Major Project of China, the project of the National Science Fund for Distinguished Young Scholars of China, the project of the National 863 High-Tech Program of China, and so on. He has published more than 100 papers and a monograph supported by the National Defense Science and Technology Book Publishing Fund.

Dr. Liu won the National Science and Technology Progress Award.