Int J Adv Manuf Technol
DOI 10.1007/s00170-011-3885-2

ORIGINAL ARTICLE

A discrete-event simulation approach with multiple-comparison procedure for stochastic resource-constrained project scheduling

Shiqi Li · Yan Jia · Junfeng Wang

Received: 5 May 2011 / Accepted: 26 December 2011

Abstract Aiming to minimize the average project duration, a discrete-event simulation (DES) approach with multiple-comparison procedure is presented to solve the stochastic resource-constrained project scheduling problem (SRCPSP). The simulation model of SRCPSP is composed of a resource management model and a project process model, where the resource management model is used to administrate resources of the project, and the project process model based on an extended-directed-graph is proposed to describe the precedence constraints and resource constraints in SRCPSP. A simplified simulation strategy based on activity scanning method is used in the simulation model to generate feasible schedules of the problem. A multiple-comparison procedure based on the common random numbers is adopted to compare the multiple scheduling alternatives obtained from the stochastic simulation model and provide more information to select the optimal scheduling alternative. The cases are given to compare with other methods for the same SRCPSP from literature and show that the simulation tool by utilizing DES with a statistical method improves the efficiency of simulation in stochastic project planning.

Keywords Discrete-event simulation · Multiple-comparison procedure · Common random numbers · Stochastic resource-constrained project scheduling

S. Li · Y. Jia · J. Wang (✉)
Department of Industrial and Manufacturing System Engineering,
School of Mechanical Science and Engineering,
Huazhong University of Science and Technology,
Hubei, Wuhan 430074, People's Republic of China
e-mail: wangjf@mail.hust.edu.cn

1 Introduction

Resource-constrained project scheduling problem (RCPSP) is an NP-hard problem and concerned with single-item or small batch production where scarce resources have to be allocated to dependent activities over time [1]. The models in this area are rich, and many problems can be considered as the type of RCPSP. For instance, the job-shop scheduling problem (JSP) is a special case of RCPSP [2–4]. Existing literature on this topic largely focuses on the deterministic project scheduling problem with fixed activity durations [5–10]. However, in many real-world applications, one or more parameters of systems tend to be stochastic, such as the arrive times and due dates of jobs in JSP [4], and the customer demand, travel times, or a combination of these in vehicle routing problems (VRP) [11]. Generally, stochastic variables of known probability distributions are utilized to model these parameters.

Similarly, in a real-world project, activities are often subject to considerable uncertainty due to many different factors [12, 13], such as activities may take more or less time than originally estimated, resources may become unavailable, materials may arrive behind schedule, workers may be absent, and weather conditions may cause severe delays. This kind of RCPSP with random activity durations belongs to stochastic resource-constrained project scheduling problem (SRCPSP). The activity durations in SRCPSP are not known in advance and usually represented as random variables which can be drawn from historical data or a probability distribution.

There are few studies on SRCPSP and most of them concentrate on the optimal scheduling algorithms, such as greedy randomized adaptive search procedure [13], branch-and-bound [14], zero–one integer programming [15], tabu search [16], constraint programming based approximate

Published online: 17 January 2012

dynamic programming (CP-ADP) [17], priority-rule-based methods and genetic algorithm [18, 19]. In the SRCPSP, since the activity durations are randomly distributed (e.g., beta, uniform and normal distributions), the corresponding project duration is also a random variable. Therefore, in order to evaluate the stochastic output performances, Stork [14], Golenko-Ginzburg and Gonik [15], and Tsai and Gemmill [16] took a user-specified number (e.g., 100) of repeating experiments for each feasible scheduling alternative obtained by their methods, and the alternative with the minimal sample mean, i.e., minimal average project duration, was selected as the best solution. Some other sampling techniques [13, 17–19] (e.g., Monte Carlo simulation) are also used to obtain high-quality solution with the minimal average project duration. All these algorithms can obtain feasible or optimal solutions for SRCPSP. However, it is difficult for them to model the stochastic and dynamic aspects of the problem, especially when a project has more activities and complicated constraints.

Discrete-event simulation (DES) as a powerful computing technique for understanding the behavior of systems can be used to describe the dynamic and stochastic perspectives of a project, and also help project managers understand the project structure in a simple and convenient way, without the need to build mathematical models. The uncertainty characteristic of the project lends itself very well to simulation applications [20]. Badiru [20] developed a computer program, named STARC, to illustrate the effectiveness of computer simulation analyses for project planning. And in the use of simulation, the activity durations are modeled using three time estimates (i.e., an optimistic time, the most likely time, and a pessimistic time) and generated randomly from a fitted beta probability distribution function. Reddy et al. [21] developed a modeling and scheduling system which coupled genetic algorithm with Petri nets to solve multi-mode and multi-resource-constrained project scheduling. But the randomicity of the activity durations which would result in random performances is ignored in the proposed method. Zhang et al. [22–24] combined DES with different methods to solve deterministic, stochastic, and fuzzy construction project scheduling problems. In the work of Zhang et al. [23], the authors proposed a simulation–optimization method integrated particle swarm optimization with DES to determine the optimal resource combination for a construction operation. The activities that have the stochastic durations are defined in triangular probability distributions, and a multiple-comparison procedure, namely MCP-MN, is adopted to compare multiple resource combination alternatives so as to select the best one which has the maximal mean productivity.

When stochastic behaviors such as random activity durations are considered in the simulation model, the corresponding output performances contain random variances [23]. So, special statistical method is required to evaluate the output performances of different scheduling alternatives. Research in SRCPSP, most literature exploits the fixed sample size to evaluate each alternative, which cannot get satisfied solutions or results in a lower efficiency. Therefore, a suitable sample size needs to be determined to avoid a large amount of experiments which may slow down the simulation efficiency.

There are two kinds of popular statistical methods in DES [25]: ranking and selection (R&S) and multiple-comparison procedures (MCPs). For R&S method, the goal is to choose the best solution from among a set of competing alternatives. For MCPs method, it can not only make a decision, but can also identify the differences between alternatives' output performances. The typical MCPs include: Rinott's procedure (procedure *R*) [26], Dudewicz and Dalal's procedure (procedure *DD*) [27], Clark and Yang's procedure (procedure *CY*) [28], as well as Nelson and Matejcik's procedure (procedure *NM*) [29]. Among these procedures, procedures *R* and *DD* adopt independent random nunmbers (IRN) for simulation. While procedures *CY* and *NM* exploit common random numbers (CRN) for simulation.

The multiple-comparison procedure MCP-MN used in Zhang et al. [23] is a kind of MCPs based on IRN which is easy to be implemented, and according to the authors the simulation–optimization under MCP-MN is more effective than the general simulation with regard to the number of experiments needed to find out the final result. It is well known that the CRN approach is a variance reduction technique which is frequently employed in system simulation, and the use of CRN can reduce the total number of replications required to achieve the desired probability of correct selection [30, 31]. For the MCPs using CRN, procedure *CY* is a conservative procedure and does not make special assumptions about the covariance induced by CRN, while procedure *NM* is based on a structural assumption (sphericity) for the covariance matrix that may or may not apply in practice, so researches tend to be on the conservative side in their computational requirements [32, 33].

In this paper, a simulation tool exploration discrete-event stochastic method is proposed to solve stochastic resource-constrained project scheduling problem. Firstly, an intuitive graphical modeling method is used to describe the problem. Then, a simplified simualtion strategy is exploited to imitate the randomicity of activities and generate the scheduling alternatives of the problem. At last, a multiple-comparison procedure is adopted to rank the different scheduling alternatives. The main contribution of this paper includes two aspects: (1) a discrete-event simulation tool is provided to generate project scheduling alternatives satisfied with the precedence constraints and resource constraints of SRCPSP; (2) a multiple-comparison procedure (procedure *CY*) based on CRN is exploited to compare the multiple scheduling alternatives obtained from the stochastic simulation so as to select the best scheduling alternative with the minimal average project duration.

The paper is organized as follows. In the next section, we describe the stochastic resource-constrained project scheduling problem. In Section 3, we introduce our simulation procedure for generating stochastic resource-constrained project scheduling alternatives. In Section 4, we illustrate the procedure *CY* to compare multiple scheduling alternatives generated from the stochastic simulation model. In Section 5, we use two project instances to describe the implementation of the simulation-based scheduling methodology and multiple alternatives comparisons based on procedure *CY*. A summary and some conclusions are given in Section 6.

2 Problem statement

The SRCPSP can be stated as follows: a project consists of n non-dummy activities and two dummy activities. The two dummy activities with zero durations and resource requirements are used to represent the start and end of the project, respectively. The non-dummy activities are subject to two kinds of constraints. The first is precedence constraints, i.e., any activity cannot start before all of its predecessors have been finished. The second is resource constraints, i.e., any activity cannot start without satisfying its resource requirements. There are K renewable resource types, and the available amount of each resource k is R_k, k=1,…, K. Each activity i (i=1, 2,…, n) has a duration time d_i which is a random number drawn from historical data or a probability distribution, a start time s_i and a finish time f_i, and requires r_{ik} units of resource k to be processed.

The aim of SRCPSP is to obtain the optimal scheduling alternative, i.e., the feasible sequence of the n activities which leads to the minimal average project duration with the known precedence constraints and resource constraints.

When using DES to understand the behavior of system, the objective and constraints of the system are replaced by discrete-event simulation model; the decision variables are the conditions the simulation is run under; and one or several of the responses produced by the simulation model is the system performance. In this paper, according to the description of SRCPSP and the characteristics of DES, there are three procedures for solving SRCPSP based on DES as shown in Fig. 1.

In Fig. 1, the procedures of DES for SRCPSP are as follows:

1. *Building simulation model*: According to the analyses of the problem, the simulation model of SRCPSP based on graphical method is built to describe the project scheduling problem.

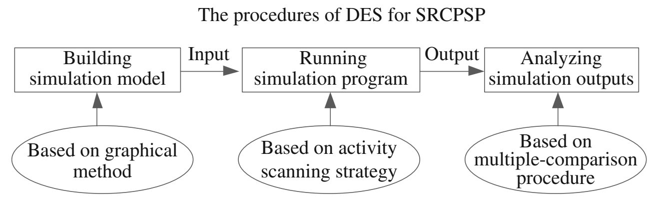


Fig. 1 The procedures of DES for SRCPSP

2. *Running simulation program*: The information of simulation model serve as the input of the simulation program. Then running the simulation program based on the activity scanning strategy, the feasible scheduling alternative of the project is generated.
3. *Analyzing simulation outputs*: The responses produced by the simulation model are analyzed based on the multiple-comparison procedure so as to select the best solution.

3 Simulation-based schedule generation for SRCPSP

3.1 The representation of simulation model for SRCPSP

1. Resource management model

   In project management, a resource pool is a set of resources available in the project, and the use of resource pool makes it easier to administrate the different resources assigned to activities. In this paper, in order to manage kinds of resources more effectively, a resource management model is proposed, where many resource pools are defined and the resources with the same type are put in a same pool which is named after the resource type. Each resource pool is described using a two-tuple as follows:

   P=<*Type*, *Policy*>, where *Type* is the type of resources (e.g., human resource) in the resource pool and is also the unique identification of the resource pool and *Policy* is the allocation rule of these resources (e.g., select activities randomly, RAN) in this pool.

   In addition, each resource in a resource pool is defined using a three-tuple as follows:

   R=<*Name*, *Count*, *Available*>, where *Name* is the resource name and is also the unique identification of the resource, *Count* is the total amount of the resource, and *Available* is the state of resource (i.e., busy or idle).

   Therefore, each particular resource in the resource management model is represented by a combination of (*P*, *R*), i.e., five-tuple<*Type*, *Policy*, *Name*, *Count*, *Available*>.

2. Project process model

In general, the project process is depicted by an activity-on-node (AON) network due to its easy and visualization. Both Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) use this method to analyze the project. However, the AON model only represents the activities and their precedence constraints in the project, but ignores the resource constraints of activities, which will only rarely be the case in a real-world project. Therefore, an extended-directed-graph (EDG) based on AON is proposed to model the resource-constrained project process. The EDG for SRCPSP can be represented as EDG=(N, A), where N is the set of nodes, and A is the set of directed arcs.

(a) Set $N=\{AEN, REN, DSN, DEN\}$ is the set of node types. There are four different types of nodes in EDG, and each type is defined as follows:

- $AEN=\{ae_1, ae_2, \ldots, ae_n\}$ is the set of non-dummy activity nodes, where n is the number of non-dummy activities in the project.
- $REN=\{re_1, re_2, \ldots, re_m\}$ is the set of resource nodes, where m is the number of activities which require resources.
- $DSN=\{ds\}$ is the dummy start node representing the start of the project.
- $DEN=\{de\}$ is the dummy end node meaning the end of the project.

And the attributes of different types of nodes are set as follows:

- *ae*=<*ID_a*, *Name_a*, *Time_a*, *Description_a*>, where *ID_a* is the unique identification of activity node, *Name_a* is the name of the node, *Time_a* is the activity duration, and *Description_a* is the job description about the activity.
- *re*=<*ID_r*, *Name_r*, *Type*, *Name*, *Requirement*>, where *ID_r* is the unique identification of resource node, *Name_r* is the name of the node, *Type* is the resource type required by an activity, *Name* is the resource name required by an activity, and *Requirement* is the number of resources assigned to an activity. Moreover, the values of *Type* and *Name* are corresponding to the definition of resource in the resource management model.
- *ds*=<*ID_ds*>, where *ID_ds* is the identification of dummy start node.
- *de*=<*ID_de*>, where *ID_de* is the identification of dummy end node.

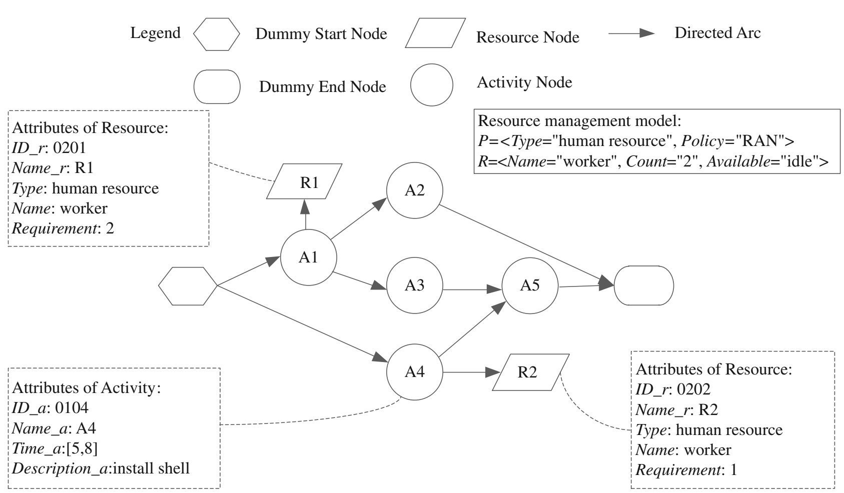
(b) Set $A=\{ar_1, ar_2, \ldots, ar_j\}$ is the set of directed arcs representing the relations between two nodes, where j is the number of directed arcs in the EDG. The attributes of directed arc is designed as follows:

ar=<*ID_ar*, *SNID*, *ENID*>, where *ID_ar* is the identification of the directed arc; *SNID*=<*ID_ds*>|<*ID_a*>is the start-node ID of the directed arc; *ENID*=<*ID_de*>|<*ID_a*>|<*ID_r*>is the end-node ID of the directed arc.

Based on the project resource management model and process model described above, an example of SRCPSP simulation model is built as shown in Fig. 2.

In Fig. 2, the resource management model contains one resource pool, and in the pool it defines two available workers which belong to the type of human resource and are allocated using RAN policy. The project process model includes five non-dummy activity nodes, two resource nodes, and two dummy activities standing for the start and end of the project, respectively. For the non-dummy activities, A1 and A4 are the starting activities, while A2 and A5 are the ending activities; the precedence relation among A1, A2, and A3 state that

Fig. 2 An example of SRCPSP simulation model



the parallel activities A2 and A3 can be activated and started when A1 has been finished; while the relation among A3, A4, and A5 indicate that A5 can be started until the parallel activities A3 and A4 both have been finished. In addition, A1 requires resource R1, while A4 needs R2. From the attributes definition of different nodes, it can be known that A4 is an activity for installing shell and its activity duration is a stochastic number in the given interval [5, 8]; the resource R1 has been defined two workers, and A4 needs one worker defined in R2. However, the total amount of the workers is two, so there is a resource competition between A1 and A4 and will be solved using the RAN policy.

3.2 Data structure of simulation model

In order to provide an easy and executable form for computer simulation, the resource management model and project process model should converted to another forms which can be implemented on computer. In this paper, the Extensible Markup Language (XML) document is used to represent and store the resource management model. In addition, two kinds of matrices are used to store the information of project process model. One is activity dependency matrix (ADM), which is used to express the activities and precedence constraints. The other is activity resource matrix (ARM), which is adopted to describe the activities and resource constraints.

The construction process of XML document corresponding to the resource management model is as follows:

1. First, build the root node (i.e., Proot) of the document, then take each resource pool (e.g., P) in the resource management model as its sub-node and add the resource type and allocation policy (i.e., <*Type*, *Policy*>) as the attributes of the sub-node.
2. Then, for each sub-node (e.g., P), take each resource in this resource pool as its sub-node and add the resource information (i.e., <*Name*, *Count*, *Available*>) as the attributes of the sub-node.

Therefore, the XML document corresponding to the resource management model in Fig. 2 is shown below.

```
<?xml version="1.0" encoding="GB2312"?>
 <Proot>
   <P Type="human resource" Policy ="RAN" >
     <R Name="worker" Count="2" Available="idle" />
   </P>
 </Proot>
```

The construction process of ADM is the mapping process of EDG to ADM. The definition of ADM is as follows:

1. The number of ADM rows and columns are both equal to the number of activity nodes in EDG.
2. The element a_{ij} of ADM is designed as follows:

$$a_{ij} = \begin{cases} 1, & \text{if activity } j \text{ is a direct precedence constraint to activity } i \\ 0, & \text{otherwise} \end{cases}$$

Hence, the ADM corresponding to the EDG in Fig. 2 is shown below.

$$\mathrm{ADM} = \begin{array}{cc} & \begin{matrix} A1 & A2 & A3 & A4 & A5 \end{matrix} \\ \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \\ A5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

The construction process of ARM is the mapping process of EDG to ARM. The definition of ARM is as follows:

1. The number of ARM rows is equal to the number of resource nodes in EDG, and the number of ARM columns is equal to the number of activity nodes in EDG.
2. The element b_{ij} of ARM is designed as follows:

$$b_{ij} = \begin{cases} 1, & \text{if activity } j \text{ needs resource } i \\ 0, & \text{otherwise.} \end{cases}$$

So, the ARM corresponding to the EDG in Fig. 2 is shown below.

$$\mathrm{ARM} = \begin{array}{cc} & \begin{matrix} A1 & A2 & A3 & A4 & A5 \end{matrix} \\ \begin{matrix} R1 \\ R2 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

3.3 A simplified simulation strategy

Simulation model can describe the SRCPSP, but cannot give a feasible solution to reflect the execution process of a real-world project. Therefore, a suitable simulation strategy is required to control the execution of simulation model so as to generate scheduling alternatives for SRCPSP.

In DES, the most common simulation strategies include event scheduling (ES), activity scanning (AS), and process interaction (PI) [34]. ES is often used to enhance PI and AS, and AS is particularly suitable to systems with interdependent components subject to complex activation conditions where many resources with distinct properties may collaborate according to highly dynamic rules [34, 35]. According to the characteristics of SRCPSP, a simplified simulation strategy based on AS is applied to the simulation model.

The simplified AS views the system as the composition of activities which are subject to specified activation conditions. During simulation advancement, the system scans the unscheduled activities and executes the activities that are satisfied with the conditions (i.e., precedence constraints and resource constraints). Therefore, with the known precedence constraints stored in ADM and resource constraints stored in ARM as well as the information in resource management model, a simplified AS simulation strategy is presented in Fig. 3 and detailed as follows:

1. Firstly, the current time (m_CT) is initialized as 0 and the records for end events (m_EEL) are initialized as empty, while the records for scanned activities (m_SL) are initialized as all the unscheduled activities in the project.
2. Check if there is any end event in m_EEL happening at m_CT, if No, go to next step; if Yes, do the following operations: release resources occupied by the activity that the current end event points to and update the available amount of the resources; delete the current end event from m_EEL.
3. Scan m_SL. Check the precedence constraints of activities in m_SL, and mark the activities that are satisfied with the precedence constraints stored in ADM.
4. Then, according to the resource allocation policy (e.g., minimum slack, MINSLK) defined in the resource management model, determine the priorities of these marked activities.
5. According to the execution priorities, check the resource constraints of these marked activities based on the information recorded in ARM. If any activity is satisfied with resource constraints, then execute this activity, including: update the available amount of resources occupied by this activity; get the activity duration d_i and compute the activity end event time (f_i=m_CT+d_i), if it is a stochastic duration, then take the expected duration as the current activity duration d_i; add activity end event to m_EEL and delete the activity from m_SL.
6. When all of the marked activities in m_SL have been scanned, the m_CT is advanced to the earliest end event time in m_EEL, and then return to step 2. This iterative process will be continued until the termination of simulation. There are two termination signals: (1) all activities have been finished and (2) the specified simulation cycle times are exceeded. After the simulation, a number of scheduling alternatives will be generated which are satisfied with the precedence constraints and resource constraints of the SRCPSP.

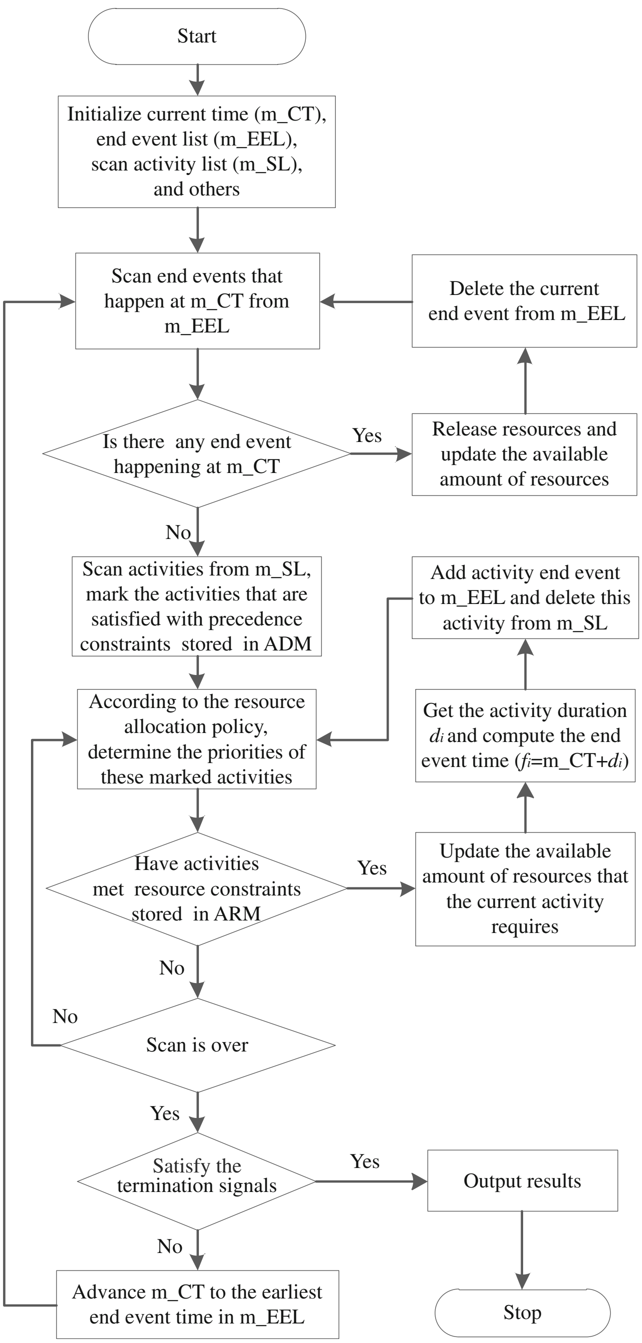


Fig. 3 A simplified AS simulation strategy for simulation model

4 Analyses of simulation outputs

4.1 Evaluation of scheduling alternative

Since the random activity durations are considered in the simulation model, the project duration resulted from the simulation is also a random variable. Therefore, given a particular feasible sequence generated by the simulation, the expected project duration should be estimated.

Suppose *PS* is a particular feasible sequence which represents the priorities of activities during the scheduling; DS_n is the set of activities which have been scheduled; $\widetilde{R}_{kt}$ is the available amount of renewable resource k at time t, and $\widetilde{R}_{kt} = R_k - \sum_{i\in A(t)} r_{ik}$, ($k$=1,2,…,$K$, and $A(t)$ is the set of activities being processed at time t). The heuristic procedure to calculate the project duration of scheduling alternative *PS* is described as follows.

Heuristic for the calculation of project duration:

1. Initialize n=1, DS_1=Ø, S=PS;
2. While there are unscheduled activities do:

   Step 1: According to the sequence of activities in S, select the activity i with the highest priority from S, and one activity duration d_i for the activity i is drawn randomly from the given distribution;

   Step 2: Get the latest finish time of predecessor activities related to activity i from DS_n, and take it as the earliest possible start time ES_i of activity i;

   Step 3: Calculate the earliest start time of activity i when it both meets the precedence constraints and resource constraints, i.e., $s_i = \min\{t|ES_i \leq \quad t, r_{ik} \leq \widetilde{R}_{k\tau}., \tau = t, \ldots, t + d_i - 1, t = 0, 1, \ldots\}$;

   Step 4: Calculate the finish time of activity i, i.e., $f_i = s_i + d_i$;

   Step 5: Remove activity i from S and update the $DS_{n+1} = DS_n \cup \{i\}$;

   Step 6: Set n=n+1;
3. End while

When the procedure is over, the project duration (i.e., f_n) of the feasible sequence *PS* is calculated. Further, when the heuristic for calculation of project duration is repeated a number of times with different sets of activity durations, and then the average project duration is reported as the expected project duration for the particular feasible sequence.

4.2 Comparison and selection of multiple scheduling alternatives

When running the simulation program with different kinds of resource allocation policies, a number of scheduling alternatives will be generated, and the aim of SRCPSP is to obtain the scheduling alternative with the minimal expected project duration. Therefore, in order to compare the multiple scheduling alternatives with random output performances and select the best one, the procedure *CY* exploiting CRN is used for comparing the multiple alternatives in this paper.

The following notation will be used throughout the procedure *CY*: i=1, 2,…, k, where k means the total number of alternatives and θ_i is the alternative i; μ_i is the expected output performance of θ_i; $Y_{i.}$ denotes the overall sample of θ_i and $\overline{Y}_{i.}$ is the sample mean; Y_{ir} represents the rth independent and identically distributed (i.i.d.) simulation output (i.e., replication) from θ_i.

For procedure *CY*, when use of CRN, the rth replication of each alternative will use the same random numbers, then the rth outputs of all alternatives $Y_{.r}$={Y_{1r}, Y_{2r},…, Y_{kr}} will be not independent, but correlated. We know that the covariance matrix of k random alternatives is as follows:

$$\sum = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} \\ & & \ddots & \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{pmatrix}$$

Where $\sigma_{ij}=Cov(\theta_i\text{-}\theta_j)=E\{[\theta_i\text{-}E(\theta_i)][\theta_j\text{-}E(\theta_j)]\}$, which represents the covariance of alternatives i and j (i, j=1, 2,…, k). The goal of CRN is to generate positive correlation between alternatives i and j, and force $\sigma_{ij}>0$. For $Var(\theta_i\text{-}\theta_j)=Var(\theta_i)+Var(\theta_j)\text{-}2\sigma_{ij}$, only if $\sigma_{ij}>0$, the variance of alternatives generated by CRN will be smaller than the variance by IRN. Therefore, the use of CRN can sharpen the comparison of two or more alternatives. In the case of ranking and selection, "sharpening" means reducing the total number of replications required to achieve the desired probability of correct selection [31].

The procedure *CY* under CRN for comparing multiple scheduling alternatives contains the following steps:

1. Specify the confidence level 1-α, indifference zone δ, number of feasible alternatives k, and the common initial sample size, i.e., the number of experiments n_0. Let $t = t_{1-[\alpha/(k-1)],n_0-1}$, which is the (1−[$\alpha$/($k$−1)])-quantile of the t distribution with n_0−1 degree of freedom.
2. Make n_0 runs of simulation experiments for each alternative i (i=1, 2,…, k), then the sample $Y_{i1}, Y_{i2}, \ldots, Y_{in_0}$ from each of the k alternatives can be obtained by using CRN across alternatives. The application of CRN to simulate the random out performances of alternatives is designed as follows:

   Step 1: Select one of the feasible scheduling alternative i from the k alternatives.

   Step 2: Calculate the project duration of this particular schedule using the heuristic procedure presented above (in section 4.1). Simultaneously, record the random vector of activity durations during the calculation, i.e., D=(d_1, d_2,…, d_n).

   Step 3: Repeat the step 2 n_0 times, then the sample Y_{ir} (r=1, 2,…, n_0) from the alternative i can be obtained, and the random time vectors D_r(r=1, 2,…, n_0) are recorded and used as CRN.

   Step 4: Then all of the other feasible schedules are simulated using the same common random activity durations D_r(r=1, 2,…, n_0).
3. Calculate the sample variances of the differences $S_{ij}{}^2 = \frac{1}{n_0-1}\sum_{r=1}^{n_0}\left(Y_{ir} - Y_{jr} - \left(\overline{Y}_{i\cdot} - \overline{Y}_{j\cdot}\right)\right)^2$, for all i≠j, i=1, 2,…, k, and $\overline{Y}_{i\cdot} = \frac{1}{n_0}\sum_{r=1}^{n_0} Y_{ir}$.

4. Calculate the second-stage sample size or number of experiments
$N = \max\left\{n_0, \max_{j \neq i} (tS_{ij}/\delta)^2\right\}$, N takes an integer value.
5. Take $N - n_0$ additional simulation experiments for each alternative i (i=1, 2,…, k) by using CRN across alternatives.
6. Calculate the overall sample means
$\overline{\overline{Y}}_{i\cdot} = \frac{1}{N}\sum_{r=1}^{N} Y_{ir}$, for i=1, 2,…, k.
7. Select the alternative with the minimal $\overline{\overline{Y}}_{i\cdot}$ as the best.
8. Simultaneously form the multiple comparisons with the best (MCB) confidence intervals

$$\mu_i - \min_{j \neq i} \mu_j \in \left[\varepsilon_i^L, \varepsilon_i^H\right] = \left[\min\left(0, \overline{\overline{Y}}_{i\cdot} - \min_{j \neq i} \overline{\overline{Y}}_{j\cdot} - \delta\right), \max\left(0, \overline{\overline{Y}}_{i\cdot} - \min_{j \neq i} \overline{\overline{Y}}_{j\cdot} + \delta\right)\right], \text{ for } i=1, 2,\ldots, k.$$

If the expected output performance (i.e., μ_i) smaller the alternative is better, the MCB confidence intervals bound the difference between the performance of each alternative and the best of the others. For example, if $\mu_i - \min_{j \neq i} \mu_j \in \left[\varepsilon_i^L, \varepsilon_i^H\right]$ is totally to the left of zero, i.e., $\varepsilon_i^H < 0$, it means that $\mu_i - \min_{j \neq i} \mu_j < 0$ without a doubt and alternative i is better than alternative j; if $\mu_i - \min_{j \neq i} \mu_j \in \left[\varepsilon_i^L, \varepsilon_i^H\right]$ is totally to the right of zero, i.e., $\varepsilon_i^L > 0$, it means that $\mu_i - \min_{j \neq i} \mu_j > 0$ without a doubt and alternative i is worse than alternative j; if $\mu_i - \min_{j \neq i} \mu_j \in \left[\varepsilon_i^L, \varepsilon_i^H\right]$ contains zero, i.e., $\varepsilon_i^L < 0$ and $\varepsilon_i^H > 0$, it means that there is no difference between alternative i and alternative j according to the current statistical data.

If the alternative i is expected to have the minimal output performance μ_i, then $\mu_i - \min_{j \neq i} \mu_j (i = 1, \ldots, k)$ indicates the true difference between the best alternative and the second. With the user-specified indifference zone δ and confidence level 1-α, it means that if we use the procedure *CY*, then the alternative with the minimal performance estimate μ_i will be selected as the best alternative, and the probability of correct selection is at least 1-α when the performance of the best is at least δ smaller than the second best, i.e.,$P\{CS\} = P\left(\mu_i - \min_{j \neq i} \mu_j \leq -\delta\right) \geq 1 - \alpha$.

5 Case studies and discussion

5.1 Case 1

The simulation tool for SRCPSP has been implemented with java. An example from Patterson set is used to demonstrate the effectiveness of the simulation method proposed in this paper, and it has also been used by Tsai and Gemmill [16] to illustrate the solution found using tabu search. This project has 11 non-dummy activities, and 3 types of renewable resources (resources A, B, and C) with total amount of 6, 7, 6, respectively. The sample project is a deterministic project, and Tsai and Gemmill solve it at first using their method. Then they modify the sample to make it stochastic, where they choose a beta distribution with the parameters (α, β) to model activity durations. Three time estimates (a, m, b), i.e., (an optimistic time estimate, the most likely time estimate, and a pessimistic time estimate), used in PERT are utilized to calculate the parameters for the beta distribution. For the three time estimates (a, m, b) of each activity, a is set to 0.8 times the deterministic activity duration, m is equal to the deterministic activity duration, and b is set to 1.5 times the deterministic activity duration. This implies that $\alpha \approx$ 2.595, $\beta \approx$4.671. The three time estimates and resource requirements of each activity are in Table 1.

5.1.1 Schedule generation based on simulation

1. Application of simulation approach to the deterministic problem

   According to the project network from [16], the project simulation model including resource management model and the EDG process model is shown in Fig. 4.

   In Fig.4a, the minimum slack first (MINSLK) resource allocation policy defined in the resource management model is also exploited by Tsai and Gemmill [16] to find a starting feasible sequence for tabu search, and the project duration for the sequence is 23 determined using their procedure. When using the same policy, the scheduling results generated by our simulation method are shown in Table 2. It can be seen that the feasible sequence with the MINSLK policy is [1, 2, 3, 7,

Table 1 The three time estimates and resource requirements of each activity [16]

Activity	Three time estimate	Resource requirements
1	2.4, 3, 4.5	3, 2, 1
2	4, 5, 7.5	2, 4, 2
3	4.8, 6, 9	3, 1, 2
4	1.6, 2, 3	4, 3, 1
5	2.4, 3, 4.5	2, 0, 3
6	2.4, 3, 4.5	1, 1, 1
7	3.2, 4, 6	3, 1, 1
8	4, 5, 7.5	2, 2, 2
9	3.2, 4, 6	3, 2, 3
10	1.6, 2, 3	4, 1, 0
11	2.4, 3, 4.5	5, 4, 2

(a) Resource management model

P=<Type="Renewable Resource", Policy ="MINSLK">
*R*1=<Name="A",Count="6",Available="idle">
*R*2=<Name="B",Count="7",Available="idle">
*R*3=<Name="C",Count="6",Available="idle">

(b) EDG process model

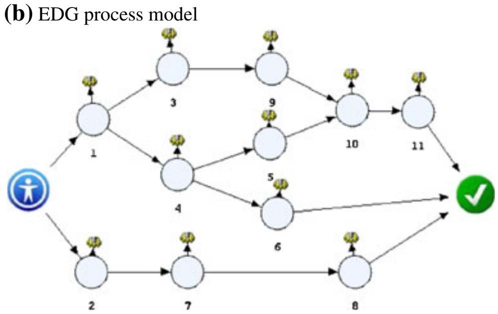


Fig. 4 Project simulation model. **a** Resource management model. **b** EDG process model

9, 8, 4, 5, 6, 10, 11], and the project duration for this sequence is 23 which is same to the result of Tsai and Gemmill.

2. Application of simulation approach to the stochastic problem

   For the stochastic version of the sample project in Fig. 4, Tsai and Gemmill [16] generate a feasible sequence by utilizing the expected duration of each activity and the MINSLK rule, and the expected project duration of the particular feasible sequence is 24.346 using a sample size of 100 with their method. When using the same resource allocation policy and sample size, our simulation method is applied to this stochastic sample project, and the results are shown in Table 3. It is obvious that the feasible sequence is [1, 2, 3, 7, 9, 8, 4, 5, 6, 10, 11], and the expected project duration is 24.242 which is nearly identical to 24.346.

From this example and the experimental results shown in this section, one can easily see that the AS-based simulation approach proposed in this paper is capable of solving the deterministic and stochastic project scheduling problem.

Table 2 Scheduling results generated by the simulation method

Activity	Start time	Duration time	Finish time
1	0	3	3
2	0	5	5
3	3	6	9
7	5	4	9
9	9	4	13
8	9	5	14
4	13	2	15
5	15	3	18
6	15	3	18
10	18	2	20
11	20	3	23

Table 3 The average duration of the sample project for the schedule determined using simulation

Activity	Mean start time	Mean duration time	Mean finish time
1	0	3.250	3.250
2	0	5.281	5.281
3	3.250	6.272	9.521
7	5.281	4.193	9.473
9	9.521	4.173	13.694
8	9.937	5.356	15.293
4	13.694	2.124	15.818
5	15.818	3.179	18.998
6	15.818	3.125	18.943
10	18.998	2.098	21.095
11	21.095	3.147	24.242

5.1.2 Scheduling alternatives comparisons

In order to describe the multiple alternatives comparisons in the stochastic project, we exploit five general resource allocation policies to generate five different scheduling alternatives for evaluation and comparison. The five rules chosen from literature [20, 36] are the MINSLK, most successor operations (MSP), shortest activity duration (SAD), great resource allocation factor (RAF) and minimum late finish time (MINLFT). The feasible sequences obtained by simulation with the different policies are shown as follows: θ_1=[1, 2, 3, 7, 9, 8, 4, 5, 6, 10, 11] with MINSLK; θ_2=[1, 2, 4, 3, 5, 6, 7, 9, 8, 10, 11] with MSP; θ_3=[1, 2, 4, 5, 6, 7, 3, 8, 9, 10, 11] with SAD; θ_4=[1, 2, 4, 5, 7, 6, 3, 8, 9, 10, 11] with RAF; and θ_5=[1, 2, 3, 7, 4, 8, 9, 6, 5, 10, 11] with MINLFT. From literature [16], the best feasible sequence is [2, 1, 4, 3, 5, 7, 9, 8, 10, 6, 11] with an average project duration of 21.706 determined using tabu search.

Table 4 Comparison results generated by procedure *CY*

Alternative θ_i	Sample size N_i	$\overline{\overline{Y}}_{i\cdot}$	$\overline{\overline{Y}}_{i\cdot}-\min_{j\neq i}\overline{\overline{Y}}_{j\cdot}$	MCB interval
1	10	23.874	2.509	[0, 3.509]
2	10	21.365	−1.751	[−2.751, 0]
3	10	24.141	2.776	[0, 3.776]
4	10	24.179	2.814	[0, 3.814]
5	10	23.116	1.751	[0, 2.751]

When using procedure CY to compare the five alternatives, the parameters are set as follows: k=5, δ=1(1 day), n_0=10, 1−α=0.95, and t=2.685. The comparison results generated by procedure CY are shown in Table 4. It is obvious that procedure CY will select alternative 2 as the best solution with the minimal sample mean 21.365 which is close to the optimal solution 21.706 determined using tabu search. The total number of replications (i.e., $\sum_{i=1}^{k} N_i$) is 50 by using CRN across alternatives, and no additional experiments are required. The MCB confidence intervals indicate that the alternatives 1, 3, 4, 5 are inferior to alternative 2, because the lower endpoints of their MCB confidence intervals are all 0.

To illustrate the impact of CRN, the covariance matrix of these five alternatives is obtained as follows. It can be seen that all σ_{ij} in the covariance matrix are positive, which means that the CRN is effective to induce positive correlation across alternatives.

$$\Sigma = \begin{pmatrix} 0.548 & 0.261 & 0.545 & 0.499 & 0.032 \\ 0.261 & 0.416 & 0.366 & 0.359 & 0.186 \\ 0.545 & 0.366 & 0.671 & 0.622 & 0.100 \\ 0.499 & 0.359 & 0.622 & 0.584 & 0.116 \\ 0.032 & 0.186 & 0.100 & 0.116 & 0.172 \end{pmatrix}$$

5.1.3 Comparisons with other statistical methods

In order to demonstrate that the procedure CY under CRN reduce the number of replications required to attain the same confidence level 1-α and whisker length δ, procedure R under IRN is used to compare the same five alternatives. For procedure R, k=5, δ=1, n_0=10, 1-α=0.95, and h=3.692 [29], and the comparison results are shown in Table 5.

From Table 5, it can be seen that procedure R also selects alternative 2 as the best solution (21.615), but it requires 85 total replications which are more than the replications of procedure CY. In procedures CY and R, the sample sizes are a function of the variances of the alternatives; the larger the variance the greater the number of replications. Because CRN can reduce the variances whenever CRN is effective, the procedure CY that exploits CRN should need significantly fewer total replications than procedure R to attain the same confidence level and whisker length, especially when a project is larger and more complicated.

Table 5 Comparison results generated by procedure R

Alternative θ_i	Sample size N_i	$\overline{\overline{Y}}_{i\cdot}$	$\overline{\overline{Y}}_{i\cdot} - \min_{j\neq i} \overline{\overline{Y}}_{j\cdot}$	MCB interval
1	10	23.874	2.259	[0, 3.259]
2	21	21.615	−2.157	[−3.157, 0]
3	16	24.575	2.960	[0, 3.960]
4	22	24.471	2.856	[0, 3.856]
5	16	23.772	2.157	[0, 3.157]

Table 6 Comparison results generated by general method

Alternative θ_i	Sample size N_i	$\overline{\overline{Y}}_{i\cdot}$	$\overline{\overline{Y}}_{i\cdot} - \min_{j\neq i} \overline{\overline{Y}}_{j\cdot}$	MCB interval
1	100	24.242	2.421	[0, 3.421]
2	100	21.821	−2.012	[−3.012, 0]
3	100	24.395	2.574	[0, 3.574]
4	100	24.645	2.824	[0, 3.824]
5	100	23.833	2.012	[0, 3.012]

For the same example, Tsai and Gemmill [16] use a general statistical method, that is, each alternative is repeated 100 times with IRN and the average project duration for the feasible sequence is reported as the expected project duration. This statistical method with the same number of replications is also exploited in this paper, and the comparison results are shown in Table 6. It can be observed that the general method also selects alternative 2 as the best solution, but it requires 500 total replications which are more than the replications of procedures CY and R.

5.2 Case 2

A more complicated case from literature [15] is also studied to demonstrate the effectiveness of the stochastic simulation method based on CRN for analyzing the larger projects. This project has 36 activities and a single renewable resource with total amount of 50. We assume that the duration of each activity is subject to beta distribution with parameters (α=2, β=3) and limited in a given interval [a, b]. The beta distribution is also considered by Golenko-Ginzburg and Gonik [15]. They do not give a feasible scheduling alternative, but only report that the minimal average project duration is 433.88 determined by a heuristic algorithm.

In our study, five alternatives generated randomly by AS-based simulation with RAN (i.e., select activities randomly) resource allocation rule are selected for comparison. For the procedures CY and R, set k=5, δ=5 (5 days), n_0=10, and 1 −α=0.95. For the general statistical method, each alternative is repeated 100 independent simulation runs. The one with the minimal sample mean is also selected as the best alternative. The comparison results of procedure CY,

Table 7 The comparison results of different statistical methods

Statistical methods	Sample size N_i	Best alternative	$\overline{\overline{Y}}_{i\cdot}$
Procedure CY	125	θ_1	430.836
Procedure R	265	θ_1	431.533
General method	500	θ_1	433.296

procedure *R* and the general statistical method for the five alternatives are shown in Table 7. It can been observed that the three methods all select the same alternative (here, θ_1=1, 5, 3, 14, 12, 2, 25, 4, 24, 7, 16, 6, 11, 23, 8, 26, 9, 10, 15, 28, 31, 32, 13, 30, 18, 17, 20, 21, 19, 34, 22, 33, 27, 29, 35, 36]) as the best solution. But procedure *CY* can generate the minimum average project duration (i.e., 430.836) and need the fewest number of replications (i.e., 125), while the general method requires the most number of replications (i.e., 500).

From the comparisons among the three statistical methods for the same SRCPSP, it can be seen that procedure *CY* using CRN is more effective than the other two methods using IRN considering the solution quality and the simulation efficiency. In addition, procedures *CY* and *R* can avoid a large amount of experiments in contrast with the general method.

6 Conclusions

A simulation-based method is proposed for solving the SRCPSP in order to select the best project scheduling alternative.

1. A resource management model is proposed to administrate all kinds of resources in the project and an extend-directed-graph based on the AON network is used to model the project process with precedence constraints and resource constraints. A simplified activity scanning strategy is adopted to generate the feasible schedules for SRCPSP. The developed simulation tool can help project engineers generate the stochastic resource-constrained project schedules conveniently and quickly.
2. A statistics method, i.e., multiple-comparison procedure, is used to carry out the comparisons of multiple scheduling alternatives with random output performances that result from the stochastic behavior of the simulation model. The MCB confidence intervals provided by the procedure *CY* enable the analyst to select the best scheduling alternative and gain insight about the best one comparison to the rest of other alternatives.
3. The experimental analyses demonstrate the effectiveness of the proposed method considering the required number of replications and the solution quality in contrast with other methods.

Currently, the method in this paper only considers the priority-rule-based scheduling policies to solve SRCPSP, the simulation integrated with other heuristic methods as well as multiple activity-execution modes are not included, and will be our further work.

Acknowledgments The authors thank the anonymous reviewers for their comments and constructive suggestions to improve this paper.

References

1. Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41
2. Márkus A, Váncza J, Kis T, Kovács A (2003) Project scheduling approach to production planning. CIRP Ann Manuf Technol 52 (1):359–362
3. Goswami M, Tiwari MK, Mukhopadhyay SK (2008) An integrated approach to solve tool-part grouping, job allocation and scheduling problems in a flexible manufacturing system. Int J Adv Manuf Technol 35(11–12):1145–1155
4. Nie L, Shao XY, Gao L, Li WD (2010) Evolving scheduling rules with gene expression programming for dynamic single-machine scheduling problems. Int J Adv Manuf Technol 50:729–747
5. Alcaraz J, Maroto C (2001) A robust genetic algorithm for resource allocation in project scheduling. Ann Oper Res 102:83–109
6. Kumanan S, Jegan Jose G, Raja K (2006) Multi-project scheduling using an heuristic and a genetic algorithm. Int J Adv Manuf Technol 31:360–366
7. Agarwal R, Tiwari MK, Mukherjee SK (2007) Artificial immune system based approach for solving resource constraint project scheduling problem. Int J Adv Manuf Technol 34:584–593
8. Shukla SK, Son YJ, Tiwari MK (2008) Fuzzy-based adaptive sample-sort simulated annealing for resource-constrained project scheduling. Int J Adv Manuf Technol 36:982–995
9. Ying KC, Lin SW, Lee ZJ (2009) Hybrid-directional planning: improving improvement heuristics for scheduling resource-constrained projects. Int J Adv Manuf Technol 41:358–366
10. Zhou YM, Guo QS, Gan RW (2009) Improved ACO algorithm for resource-constrained project scheduling problem. 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, pp 358-365
11. Nagalakshmi MR, Tripathi M, Shukla N, Tiwari MK (2009) Vehicle routing problem with stochastic demand (VRPSD): optimisation by neighbourhood search embedded adaptive ant algorithm (ns-AAA). Int J Comp Aided Engineering Technol 1(3):300–321
12. Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165:289–306
13. Ballestín F, Leus R (2009) Resource-constrained project scheduling for timely project completion with stochastic activity durations. Prod Oper Manag 18(4):459–474
14. Stork F (2000) Branch-and-bound algorithms for stochastic resource-constrained project scheduling. Research Report No. 702/2000, Technische Universität Berlin
15. Golenko-Ginzburg D, Gonik A (1997) Stochastic network project scheduling with non-consumable limited resources. Int J Prod Econ 48:29–37
16. Tsai YW, Gemmill DD (1998) Using tabu search to schedule activities of stochastic resource-constrained projects. Eur J Oper Res 111:129–141
17. Li HT (2009) Constraint programming based approximate dynamic programming for deterministic and stochastic resource-constrained project scheduling. Project Final Report, 2009. University of Missouri, St. Louis
18. Ballestín F (2007) When it is worthwhile to work with the stochastic RCPSP? J Sched 10:153–166
19. Ashtiani B, Leus R, Aryanezhad MB (2011) New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing. J Sched 14(2):157–171
20. Badiru AB (1991) A simulation approach to PERT network analysis. Simulation 57:245–255
21. Reddy JP, Kumanan S, Chetty OVK (2001) Application of Petri nets and a genetic algorithm to multi-mode multi-resource constrained project scheduling. Int J Adv Manuf Technol 17:305–314

22. Zhang H, Li H (2004) Simulation-based optimization for dynamic resource allocation. Autom Constr 13:409–420
23. Zhang H, Tam CM, Li H, Shi JJ (2006) Particle swarm optimization-supported simulation for construction operations. J Constr Eng Manag 132(12):1267–1274
24. Zhang H, Li H, Tam CM (2004) Fuzzy discrete-event simulation for modeling uncertain activity duration. Eng Constr Archit Manag 11(6):426–437
25. Swisher JR, Jacobson SH (1999) A survey of ranking, selection, and multiple comparison procedures for discrete-event simulation. Proceedings of the 1999 Winter Simulation Conference P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds, pp 492–501
26. Rinott Y (1978) On two-stage selection procedures and related probability-inequalities. Comm Stat-Theory and Methods 7(8):799–811
27. Dudewicz EJ, Dalal SR (1975) Allocation of observations in ranking and selection with unequal variances. Indian J Stat 37B (1):28–78
28. Clark GM, Yang WN (1986) A Bonferroni selection procedure when using common random numbers with unknown variances. ACM, New York, pp 313–315, Proceedings of the 1986 Winter Simulation Conference
29. Nelson BL, Matejcik FJ (1995) Using common random numbers for indifference-zone selection and multiple comparisons in simulation. Manag Sci 41(12):1935–1945
30. Heikes RG, Montgomery DC, Rardin RL (1976) Using common random numbers in simulation experiments—an approach to statistical analysis. Simulation 27:81–85
31. Nelson BL, Swann J, Goldsman D, Song W (2001) Simple procedures for selecting the best simulated system when the number of alternatives is large. Oper Res 49(6):950–963
32. Chick SE, Inoue K (2001) New procedures to select the best simulated system using common random numbers. Manag Sci 47 (8):1133–1149
33. Fu MC, Hu JQ, Chen CH, Xiong XP (2007) Simulation allocation for determining the best design in the presence of correlated sampling. INFORMS J Comput 19(1):101–111
34. Hooper JW (1986) Strategy-related characteristics of discrete-event languages and models. Simulation 46:153–159
35. Martinez JC, Ioannou PG (1999) General-purpose systems for effective construction simulation. J Constr Eng Manag 125(4):265–276
36. Davis EW, Patterson JH (1975) A comparison of heuristic and optimum solutions in resource-constrained project scheduling. Manag Sci 21(8):944–955