SCIENTIA IRANICA

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

Optimizing insuring critical path problem under uncertainty based on GP-BPSO algorithm

Z. Li and B. Li[*]

College of Management and Economics, Tianjin University, Tianjin 300072, China.

Received 7 January 2017; received in revised form 16 July 2017; accepted 7 November 2017

KEYWORDS
Insuring critical path;
Bi-level fuzzy random programming;
Hybrid algorithm;
Dynamic programming method;
Task duration;
Project management problem;
Sample average approximation.

Abstract. This study considers a novel class of bi-level fuzzy random programming problems of insuring critical path. In this study, duration of each task is considered as a fuzzy random variable and follows the known possibility and probability distributions. Because an effective way to directly solve the problem does not exist, we first reduce the chance constraint to two equivalent random subproblems under two different kinds of risk attitudes. Then, we use Sample Average Approximation (SAA) method for reformulating the equivalent random programming subproblems as the approximation problems. Since the approximation problems are also difficult to solve, we explore a hybrid Genotype Phenotype Binary Particle Swarm Optimization algorithm (GP-BPSO) for resolving two equivalent subproblems, in which Dynamic Programming Method (DPM) is used for finding the solution in the lower-level programming. At last, a series of simulation examples are provided for demonstrating the validity of the hybrid GP-BPSO compared with the hybrid BPSO algorithm.

1. Introduction

Critical path is one of the most important conceptions of operations research, which controls project time and specifies the most critical activities. In the past few years, critical path has been researched broadly in the pure and applied areas of project management. Joseph et al. [1] gave the conception of critical path, and determined the distance between the initial node and the target node. In order to find the longest path, Chen et al. [2] proposed an efficient method to analyze flow time of each task duration with two types of time constraints. Guerriero and Talarico [3] presented a conventional approach to solve network problems under three kinds of time constraints in certain circumstances.

Besides the research on deterministic critical path problem, another research area deals with the randomness of task duration. For instance, Möhring [4] minimized the resource costs for a project with each task duration being uncertain and the project was subjected to a given deadline for completion. Mitchell and Klastorin [5] developed an effective algorithm for compressing general networks with stochastic activity times. Shen et al. [6] studied a series of random optimization problems of insuring critical path. Besides this, the authors suggested decomposition approaches to settle the problems for two types of penalty functions. Goh and Hall [7] considered projects with uncertain task durations and handled the robust optimization problem with the conditional value at risk. Both Li et al. [8] and Li et al. [9] considered novel kinds of two-stage

[*]. *Corresponding author. Tel.: +86 13512495792*
E-mail addresses: lizhenhong6688@163.com (Z. Li); libo0410@tju.edu.cn (B. Li)

doi: 10.24200/sci.2017.4587

random insuring critical path problems. Since decision makers had risk aversion behavior, they used the value at risk criterion and the minimum risk criterion for the objectives, respectively. Li and Womer [10] developed valid approximate approaches with respect to the rollout policy for the project scheduling problems with uncertain activity times and resource constraints.

With the development of the fuzzy set theory proposed by Zadeh [11], fuzzy critical path problems have been considered by several researchers, such as those of references [12-20]. Among them, Chen and Hsueh [12] explored an ordinary method in order to resolve the problem of critical path method if task durations were considered as fuzzy numbers. Zammori et al. [13] proposed an innovative framework in order to take additional important fuzzy parameters into account. Amiri and Golozari [14] developed an effective method, which considered four types of factors to determine the longest path, in which the parameter of each factor was characterized by a fuzzy number. Zareei et al. [15] gave a novel project scheduling algorithm for solving fuzzy longest path issue, in which task duration was considered as a fuzzy number. Li and Dai [16] modeled a risk aversion problem of insuring critical path in fuzzy decision systems. Kaur and Kumar [17] presented a novel approach to finding the best solution in the fuzzy longest path problem. Jayagowri and Geetharamani [18] presented a special way to find the fuzzy longest path without converting the fuzzy activity times to crisp numbers. Pelikan et al. [19] utilized the fuzzy approach to solving optimization problem of project duration, critical path, and project resource allocation. Sireesha and Shankar [20] presented a novel approach to finding the fuzzy total floats and the longest path; their method could not find fuzzy earliest and latest times.

Nevertheless, in the uncertain hybrid environment, there is fuzziness and randomness concurrence in uncertain decision systems. For this reason, Liu and Liu [21] explored kinds of measure criteria with regard to fuzzy random variables, which had been used successfully in extensive meaningful mathematical programming problems, including the convergence of the approximation problem [22], data envelopment analysis [23], project scheduling problems [24], hub location problem [25,26], hazardous materials transportation problem [27], production-inventory system problem [28], equilibrium standby redundancy optimization problem [29], and so on. For insuring critical path problem, the project manager cannot acquire the complete information of every task duration in the project management systems, yet some experts may partially estimate probability distribution functions of the task durations. Under such circumstances, the stochastic task durations are known for fuzzy information. Therefore, for project managers, the randomness and fuzziness of the data are often mixed together in each task duration. However, to our knowledge, no research has considered the insuring critical path problem in fuzzy random environments. This motivated us to extend the application of the fuzziness and randomness to optimize the insuring critical path problem.

The aim of this paper is to employ equilibrium chance [30] as a modeling criterion to handle insuring critical path problem, in which each task duration is considered as fuzzy random variable. That is, a bi-level fuzzy random programming problem for insuring critical path is proposed, in which a penalty function is adopted in the upper-level programming; it is non-decreasing and characterized by convexity or non-convexity. Furthermore, each task duration in the lower-level programming is considered as a fuzzy random variable. In fact, because it is hard to compute the chance function in this problem, we first reduce each chance constraint to its equivalent probabilistic constraint for triangular fuzzy random task duration. Subsequently, we employ the SAA [29,31] to reformulate the equivalent probabilistic constraints as their approximation ones. Since the approximation problems are also difficult to solve, we will not compute approximation programming problems by traditional optimization methods. In this study, we adopt a modified Binary Particle Swarm Optimization (BPSO) algorithm to settle the approximation problem of insuring critical path.

The BPSO algorithm, proposed by two experts [32] in 1997, has pullulated in the literature [33-38] and the modified and improved BPSOs are successfully employed for the substantial programming problems. For example, Lee et al. [33] studied amendments and improvements of the original BPSO. Wang and Watada [34] proposed a hybrid PSO algorithm for fuzzy random facility location problem with value at risk. Beheshti et al. [35] introduced memetic BPSO to solve discrete optimization problems. Li et al. [36] used chaotic BPSO for stochastic optimal operation of microgrid. Liu et al. [37] analyzed the inertia weight parameter of BPSO. Jiang et al. [38] presented a new hybrid optimization method with wavelet mutation. Motivated by the aforementioned work, our article proposes a novel hybrid method, which combines SAA, DPM, BPSO, and genotype-phenotype mechanism, to compute the fuzzy random programming problem, in which DPM is used for finding objective of the lower-level programming. For the above hybrid resolution method, the phenotype presents position information and the genotype presents direction information, which are applied to encode schemes. They clearly keep the searching information in every step and are compared with each other for obtaining the best solution in the search space. These features give us the motivation to incorporate the genotype-phenotype mechanism into the BPSO in order to resolve the problem of insuring critical path. At last, a series of simulation examples are provided for demonstrating the validity of hybrid GP-BPSO method.

The remainder of this study is as follows. The second section gives a novel bi-level fuzzy random programming problem of insuring critical path. In the third section, the original problem is converted into two equivalent random optimization models. In the fourth section, the stochastic programming problems can be transformed into an approximate 0-1 mixed integer programming problem. The fifth section proposes a hybrid algorithm for resolving the approximate problems. The sixth section provides a series of simulated examples to testify the efficiency of the hybrid method. At last, conclusions are drawn in the seventh section.

2. Problem description

This paper studies the insuring critical path problem in complex project management. The project will be subjected to a given completion date. If the project is not completed within the deadline time, there is penalty for delay. To ensure completion time for the project, the project manager insures some critical arcs by prehiring additional labor or paying additional money. For insuring critical path problem, some experts partially know the probability distribution functions of the task durations by estimation. In other words, the random task durations are known with fuzzy information, so every task duration is described by fuzzy random variable. Based on the above consideration, a bi-level fuzzy random programming model of insuring critical path is established in this section. In this way, some notations and parameters in the network are defined in the following:

i, j	Node ordering, $i \in N$, $j \in N$;
N	Node set, $N = \{0, 1, 2, \cdots, n\}$;
A	Arc set, $A \subset N \times N$, where A is composed of two nodes that satisfy topological order sequence, that is, if $i < j$, then $(i, j) \in A$;
$G(N, A)$	Digraph to represent any completed project;
$FS(i)$	The right adjacent node set, $\forall\, i \in N$, $FS(i) = \{j \mid (i, j) \in A\}$;
$RS(i)$	The left adjacent node set, $\forall\, i \in N$, $RS(i) = \{j \mid (j, i) \in A\}$;
ω	One case from Ω, where Ω stands for the set of possible limited cases;
c_{ij}	The cost of insuring task duration without delay on the arc (i, j);
d_{ij}^{ω}	Task duration without insuring arc (i, j) under scenario ω, fuzzy random variables;
g_{ij}^{ω}	Task duration with insuring arc (i, j) under scenario ω, fuzzy random variables;
Θ	The non-decreasing penalty function for delay deadline;
$Q(x, \omega)$	The critical path length of a scenario ω;
α	A preselected risk-level parameter, where $0 < \alpha \leq 1$;
x	A decision vector, where the component part, x_{ij}, takes 1 if the project manager insures arc (i, j); or else, if the project manager does not insure arc (i, j), then $x_{ij} = 0$;
y_{ij}^{ω}	Binary decision variables, y_{ij}^{ω}, takes 1 if arc (i, j) belongs to a certain critical path under scenario ω, otherwise $y_{ij}^{\omega} = 0$.

Based on the above-mentioned notations and parameters, a bi-level programming for fuzzy random insuring critical path can be constructed as follows:

$$
\begin{aligned}
&\min \quad \varphi \\
&\text{s.t. :} \\
&\qquad \Pr\left\{\omega \in \Omega \middle| Cr\left\{\sum_{(i,j)\in A} c_{ij} x_{ij} + \Theta(Q(x,\omega)) \leq \varphi\right\} \geq 1-\alpha\right\} \geq 1-\alpha \\
&\qquad x_{ij} \in \{0,1\}, \qquad \forall\, (i,j) \in A,
\end{aligned}
\tag{1}
$$

where φ is the minimum objective value, which is a threshold of the total cost consisting of insuring and penalty, Cr represents credibility measure in the reference [39], Pr is probability measure, and $Q(x, \omega)$ represents the optimal value in the lower-level programming:

$$
\begin{aligned}
&\max \quad \sum_{(i,j)\in A} (d_{ij}^{\omega} - (d_{ij}^{\omega} - g_{ij}^{\omega}) x_{ij}) y_{ij}^{\omega} \\
&\text{s.t. :} \\
&\qquad \sum_{j \in FS(0)} y_{0j}^{\omega} = 1 \\
&\qquad \sum_{j \in FS(i)} y_{ij}^{\omega} - \sum_{l \in RS(i)} y_{li}^{\omega} = 0, \qquad \forall\, i \in N \backslash \{0, n\} \\
&\qquad y_{ij}^{\omega} \in \{0,1\}, \qquad \forall\, (i,j) \in A.
\end{aligned}
\tag{2}
$$

In upper-level programming, the first constraint includes two costs: the total cost of insuring arcs, $\sum_{(i,j)\in A} c_{ij}x_{ij}$, and a penalty fee, $\Theta(Q(x,\omega))$, in scenario ω. With a preselected risk-level parameter, $\alpha \in (0,1]$, the first constraint together with the objective function, φ, represents a $(1-\alpha)$-critical threshold point of the minimum cost objective. In addition, the second constraint gives a series of 0-1 decision variables. The objective of the lower-level programming (2) is obtaining the finish time for the project. The first constraint ensures the unique starting arc of the critical path, the second constraint ensures the balance of arc numbers in the critical path, and the third constraint gives a series of 0-1 decision variables.

The upper-level programming (1) is a class of 0-1 integer fuzzy random problems, which is extremely difficult to solve directly. One of the reasons is that for a given decision, x_{ij}, the first constraint contains fuzziness and randomness simultaneously. Besides this, $Q(x,\xi(\omega))$ is unknown. To overcome these difficulties, we reduce the first constraint from Model (1) to the equivalent stochastic ones by using triangular fuzzy random variables in the next section.

3. Equivalent conversion problem

This section first discusses the transformational model of the upper-level insuring critical path model (1). We suppose that uninsured or insured task duration for each arc is known as a triangular fuzzy random variable. Accordingly, the first constraint in programming (1) can be simplified to a random equivalence transformation form. For simplicity of notation, let $H(\omega)=\Theta(Q(x,\omega))$.

For each $\omega\in\Omega$ and any $(i,j)\in A$, let d_{ij}^{ω} and g_{ij}^{ω} be the triangular fuzzy random variables, $d_{ij}^{\omega}=(d_{ij}^{\omega 1},d_{ij}^{\omega 2},d_{ij}^{\omega 3})$ and $g_{ij}^{\omega}=(g_{ij}^{\omega 1},g_{ij}^{\omega 2},g_{ij}^{\omega 3})$ be mutually independent fuzzy variables, and $d_{ij}^{\omega 1},d_{ij}^{\omega 2},d_{ij}^{\omega 3},g_{ij}^{\omega 1},g_{ij}^{\omega 2}$, and $g_{ij}^{\omega 3}$ be random variables. According to [25,40], we can transform the first inequality constraint of problem (1) into two subproblems as follows:

(i) If $0<\alpha\le\frac{1}{2}$, then:

$$\Pr\left\{\omega\in\Omega\,|Cr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+H(\omega)\le\varphi\right\}\ge 1-\alpha\right\}\ge 1-\alpha, \tag{3}$$

can be converted into the following form:

$$\Pr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+2\alpha H(\omega 2)+(1-2\alpha)H(\omega 3)\le\varphi\right\}\ge 1-\alpha,$$

$$\forall\,(i,j)\in A. \tag{4}$$

(ii) If $\frac{1}{2}<\alpha\le 1$, then:

$$\Pr\left\{\omega\in\Omega\,|Cr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+H(\omega)\le\varphi\right\}\ge 1-\alpha\right\}\ge 1-\alpha, \tag{5}$$

can be converted into the following form:

$$\Pr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+(2\alpha-1)H(\omega 1)+(2-2\alpha)H(\omega 2)\le\varphi\right\}\ge 1-\alpha,$$

$$\forall\,(i,j)\in A. \tag{6}$$

From the discussion above, we can summarize the translation results of Model (1) as follows.

If $0<\alpha\le\frac{1}{2}$, then we could transform the original problem (1) into a stochastic programming problem as follows:

$$\min\quad\varphi$$

$$\text{s.t.}:$$

$$\Pr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+2\alpha H(\omega 2)+(1-2\alpha)H(\omega 3)\le\varphi\right\}\ge 1-\alpha$$

$$x_{ij}\in\{0,1\},\qquad\forall\,(i,j)\in A \tag{7}$$

If $\frac{1}{2}<\alpha\le 1$, then we could convert the original problem (1) into a stochastic programming problem as follows:

$$\min\quad\varphi$$

$$\text{s.t.}:$$

$$\Pr\left\{\sum_{(i,j)\in A}c_{ij}x_{ij}+(2\alpha-1)H(\omega 1)+(2-2\alpha)H(\omega 2)\le\varphi\right\}\ge 1-\alpha$$

$$x_{ij}\in\{0,1\},\qquad\forall\,(i,j)\in A \tag{8}$$

Problems (7) and (8) are also extremely difficult to simultaneously solve. The reason may be that for a given upper-level decision, x_{ij}, the probabilistic constraint of the upper-level model is hard to compute, since it requires a multi-dimensional integration. Furthermore, to solve this problem, we still require the conversion of the probabilistic constraints to their approximation problems.

4. Solution method

In practical decision making, most of the decision makers may select the behavior with high risk aversion, that is, $0 < \alpha \leq \frac{1}{2}$. Thus, we convert model (7) into its approximate programming problem.

When the risk-level parameter α satisfies the condition $0 < \alpha \leq \frac{1}{2}$, the following probabilistic constraint from model (7) is considered:

$$\Pr\left\{ \sum_{(i,j)\in A} c_{ij}x_{ij} + 2\alpha H(\omega 2) + (1-2\alpha)H(\omega 3) \leq \varphi \right\} \geq 1-\alpha. \tag{9}$$

Further, we consider the case when the task durations of arcs $d_{ij}^{\omega 1}$, $d_{ij}^{\omega 2}$, $d_{ij}^{\omega 3}$, $g_{ij}^{\omega 1}$, $g_{ij}^{\omega 2}$, and $g_{ij}^{\omega 3}$ are general discrete random variables. For convenience, we give the probability distribution $p_k > 0$, $\sum_{k=1}^{K} p_k = 1$ $(k = 1, 2, \cdots, K)$.

At this point, we introduce a constant M that is big enough, i.e.:

$$\sum_{(i,j)\in A} c_{ij}x_{ij} + 2\alpha H(\omega 2) + (1-2\alpha)H(\omega 3) - M \leq \varphi. \tag{10}$$

Subsequently, by introducing a vector z of binary variables, we may calculate the number of satisfying constraints. The vector is composed of some component z_k, which is 1 if the corresponding constraint is not satisfied and 0 otherwise.

Based on the definition of probability measure [41,42], the SAA method can be employed for the probability constraint in model (7). Then, we can transform model (7) into approximate 0-1 mixed-integer programming problem:

$$\begin{aligned} &\min \quad \varphi \\ &\text{s.t.}: \\ &\qquad \sum_{(i,j)\in A} c_{ij}x_{ij} + 2\alpha H(\omega 2) + (1-2\alpha)H(\omega 3) \\ &\qquad -Mz_k \leq \varphi \\ &\qquad \sum_{k=1}^{K} p_k z_k \leq \alpha \\ &\qquad z_k \in \{0,1\}, \qquad k = 1, 2, \cdots, K \\ &\qquad x_{ij} \in \{0,1\}, \qquad \forall\, (i,j) \in A. \end{aligned} \tag{11}$$

In upper-level programming problem, $Q(x, \omega)$ is the time limit for a project and the best value for the lower-level programming:

$$\begin{aligned} &\max \quad \sum_{(i,j)\in A} (d_{ij}^{\omega} - (d_{ij}^{\omega} - g_{ij}^{\omega})x_{ij})y_{ij}^{\omega} \\ &\text{s.t.}: \\ &\qquad \sum_{j\in FS(0)} y_{0j}^{\omega} = 1 \\ &\qquad \sum_{j\in FS(i)} y_{ij}^{\omega} - \sum_{l\in RS(i)} y_{li}^{\omega} = 0, \qquad \forall\, i \in N\backslash\{0, n\} \\ &\qquad y_{ij}^{\omega} \in \{0,1\}, \qquad \forall\, (i,j) \in A. \end{aligned} \tag{12}$$

The upper-level programming (11) is a bi-level 0-1 mixed-integer programming model, in which the upper-level penalty function, $H(\xi_k)$, is unknown. Hence, model (11) cannot be resolved by traditional optimization method. In the next section, we will introduce an effective method to seek the time limit of a project for the lower-level programming (12), which is important for the solution to Problem (11).

5. The hybrid GP-BPSO algorithm

In this section, we use dynamic programming to compute the lower-level programming in the first subsection; then, we adopt the hybrid method to resolve the approximate upper-level programming by some simulation experiments in the second subsection.

5.1. Computing critical path

For solving the approximate model, by analyzing the characteristics of the lower-level programming, we adopt dynamic programming [43] to refresh objective of the lower-level programming based on a concrete computational expression as follows:

$$h(j) = \max_{(i,j)\in A} \left\{ h(i) + (q_j^i - (q_j^i - p_j^i)x_j^i) \right\}, \tag{13}$$

in which h is the length of computing critical path; for arc (i,j), q_j^i and p_j^i are task durations without and with insuring the arc, respectively; and x_j^i takes 1 if the project manager insures the arc, or else if the project manager does not insure the arc, then $x_j^i = 0$.

5.2. A hybrid GP-BPSO method

In 1997, two experts [32] turned standard continuous Particle Swarm Optimization (PSO) algorithm [44] to binary spaces. However, BPSO is a variant of the continuous PSO, which has not been studied much. Based on BPSO simulation method, we explore a novel revised GP-BPSO method. In the following, we show the concrete operating procedures.

5.2.1. Encoding representation of solution

In the algorithm, the components of the vector $x = (x_{ij})_{1\times m}$, consisting of 0 and 1 as a group of particles, are used for expressing a solution to problem (11) above, in which m represents the number of arcs. Each gene $x_{ij} = 1$ reveals that arc (i, j) is protected from delay; conversely, each gene, $x_{ij} = 0$, shows that arc (i, j) is not protected from delay time.

5.2.2. Initialization value

For the lower-level programming (11), we randomly generate a series of phenotypes from the $\{0, 1\}^m$, forming a position vector $x_p = (x_{p,1}, x_{p,2}, \cdots, x_{p,m})$. If a randomly generated number in the $(0, 1)$ interval of uniform distribution is larger than 0.5, then set it to 1; otherwise, set it to 0. Repeating this process for P_{size} times, P_{size} groups of the original particles consisting of both the phenotype particles and the genotype particles $(x_{p,1}, x_{g,1}), (x_{p,2}, x_{g,2}), \cdots, (x_{p_{P_{\text{size}}},m}, x_{g_{P_{\text{size}}},m})$ can be produced.

5.2.3. Evaluation

Let $\text{fit}(\cdot) = -\varphi$ be the fitness function. Thus, the particle with the better target is assessed by the higher fitness.

5.2.4. Renewal process of velocities of GP particles

For this operation, we need to seek the current best particle, $P_{\text{best},i} (i = 1, 2, \cdots, P_{\text{size}})$, by the end of a certain phenotype particle, $x_{p,i}$. As the renewal population of phenotype particles increases, the global best particle, G_{best}, can be produced until the end of the population of the entire phenotypes. At this time, given each i, each genotype particle $x_{g,i}$, each phenotype particle, $x_{p,i}$, and each velocity vector, $v_{i,d}$ are considered by the following regulations for renewal, respectively:

$$x_{g,i} = x_{g,i} + v_{i,d}, \tag{14}$$

$$x_{p,i,j} = \begin{cases} 1, & \text{if rand } () < S(x_{g,i,j}), \\ 0, & \text{if rand } () \geq S(x_{g,i,j}), \end{cases} \tag{15}$$

$$\begin{aligned} v_{i,d} =& w \cdot v_{i,d} + c_1 \cdot \text{rand}() \cdot (P_{\text{best},i} - x_{p,i}) \\ &+ c_2 \cdot \text{rand}() \cdot (G_{\text{best},i} - x_{p,i}), \end{aligned} \tag{16}$$

in which w is a coefficient; both c_1 and c_2 are rates; rand() is randomly generated in the interval $(0, 1)$ of uniform distribution; let $S(x) = 1/(1 + e^{-x})$; $x_{p,i,j}$ and $x_{g,i,j}$, respectively, represent the components of the vectors $x_{p,i}$ and $x_{g,i}$. Through the above process, a new generation of phenotype particles can be found.

Based on the description above, a brief outline of the hybrid algorithm to resolve the approximate problem (11)-(12) is the following:

Step 0: Install five parameters of c_1, c_2, w, P_{size}, and $v_{\max}$;

Step 1: Initialize particles;

Step 2: Renew particles;

Step 3: Solve lower-level model (12) subject to expression (13);

Step 4: Compute fitness function of each phenotype particle;

Step 5: Renew $P_{\text{best},i}$ from each renewal particle of phenotype, and renew G_{best} up to the end of the phenotype population;

Step 6: Duplicate the second step to the fifth step up to the given maximum generation number;

Step 7: Serve particle G_{best} for the approximate best solution.

6. Numerical experiments

In this section, we implement some numerical simulation examples for the above approximate problem in order to illustrate the hybrid GP-BPSO algorithm. The parameters in the BPSO part have been selected according to the referred algorithm papers [34,35,38], and the parameters in the GP part have been selected according to the referred algorithm papers [9,33]. All the algorithms based on the C++ programming language are encoded by a Windows 8 system (ThinkPad with Intel(R) Core(TM) i5-3337U 1.80GHZ CPU and 4.00GB RAM).

First, a given project for the above fuzzy random problem is shown in Figure 1. In this fuzzy random problem, we suppose $M = 10^5$. Besides this, the penalty function in the above model is the following:

$$\Theta(t) = \begin{cases} 0, & 0 < t \leq 940, \\ 120 + (t - 940)^2/400, & 940 < t \leq 950, \\ 240 + \sqrt{t - 950}/50, & 950 < t \leq 960, \\ 360 + (t - 960)^2/200, & t > 960. \end{cases} \tag{17}$$

On every arc $(i, j) \in A$, the value of each task duration satisfies the following condition:

$$d_{ij}^k \sim (a, b, c), \quad a \sim U[80, 100],$$

$$b \sim U[180, 200], \quad c \sim U[280, 300].$$

and g_{ij}^k for arc $(i,j) \in A$ are all generated from the uniformly distributed $[0.5d_{ij}^k, 0.8d_{ij}^k]$. At the same time, the cost c_{ij} is obtained by obeying uniform distribution in the interval $[60, 80]$. At last, all d_{ij}^k, g_{ij}^k, and c_{ij} take integer values.

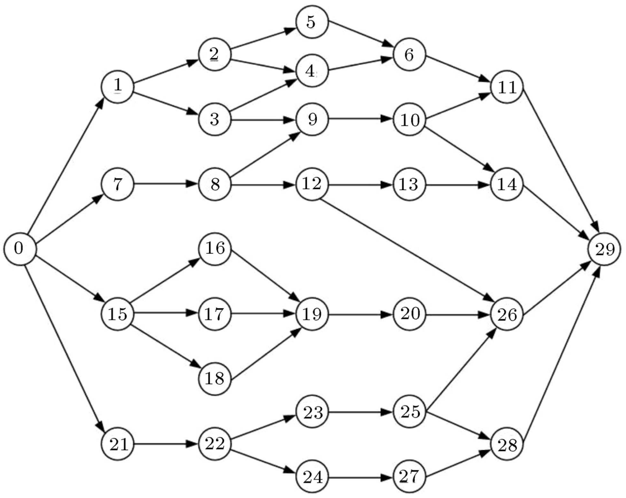

Figure 1. A project in which each circle represents a node, each line with an arrow represents an arc, and each number represents the number of the node.

For our hybrid GP-BPSO approach, we let the population size be $P_{\text{size}} = 30$, the maximum velocity be $v_{\max} = 2$, and the learning rates be $c_1 = c_2 = 2$. The weight w decreases between 0.9 and 0.4, which is expressed as follows:

$$w = \frac{1}{2GEN}(GEN - gen) + \frac{2}{5}, \tag{18}$$

in which gen and GEN are iteration times for the current and maximum cases, respectively.

For a series of insuring critical path problems, Table 1 outlines all computations through the proposed GP-BPSO method above for a project containing 30 nodes and 42 arcs. Column 1 provides the number of sample sizes; column 2 reports decision makers' four different kinds of risk attitude; column 3 addresses the arcs for insuring according to their topological order in Figure 1; and column 4 shows the threshold values of the allowable total cost.

To fully evaluate practical efficacy of the method proposed above, we report the comparison results between the hybrid GP-BPSO algorithm and the hybrid BPSO algorithm, which are shown in Table 2 and Figures 2-5.

Table 1. Simulation outcome of our modified algorithm ($GEN = 300$).

K	α	Optimal solution	φ
1000	0.20	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1)	935
	0.15	(1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0)	1034
	0.10	(0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1)	1073
	0.05	(0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0)	1190
3000	0.20	(1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1)	944
	0.15	(1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0)	1035
	0.10	(1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0)	1074
	0.05	(1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0)	1191
5000	0.20	(1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1)	947
	0.15	(1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0)	1039
	0.10	(1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1)	1076
	0.05	(1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1)	1194

Table 2. Comparison results ($GEN = 300$).

α		K		
		1000	**3000**	**5000**
0.05	Hybrid GP-BPSO	1190	1191	1194
	hybrid BPSO	1203	1204	1206
0.10	Hybrid GP-BPSO	1073	1074	1076
	Hybrid BPSO	1123	1126	1127
0.15	Hybrid GP-BPSO	1034	1035	1039
	Hybrid BPSO	1062	1062	1066
0.20	Hybrid GP-BPSO	935	944	947
	Hybrid BPSO	983	984	985

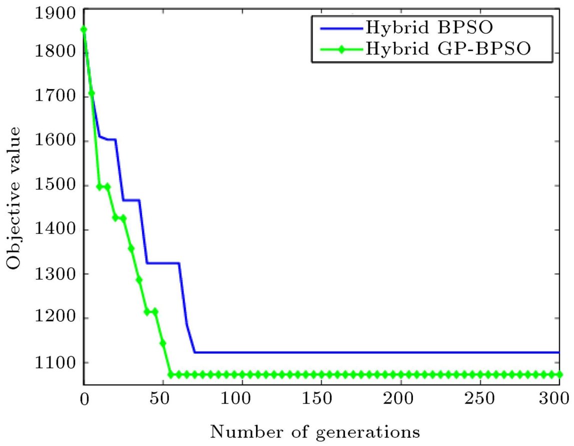

Figure 2. Performance comparison with $GEN = 300$, $\alpha = 0.10$, and $K = 1000$.

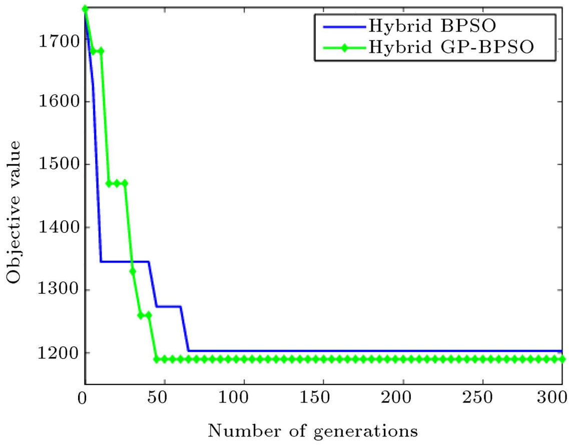

Figure 3. Performance comparison with $GEN = 300$, $\alpha = 0.05$, and $K = 1000$.

7. Conclusions

This paper proposed a new bi-level fuzzy random programming model for insuring critical path in which a penalty function was adopted in the upper-level programming. Also, it was non-decreasing and characterized by convexity or non-convexity. Besides these, each task duration in the lower-level programming was assumed to be a fuzzy random variable. For the general task duration distributions, we could not resolve the proposed model by traditional programming methods. To avoid this difficulty, we first transformed the proposed insuring critical path problem into the equivalent random programming subproblems. Subsequently, the SAA method was used for rewriting the equivalent submodels into their approximation problems.

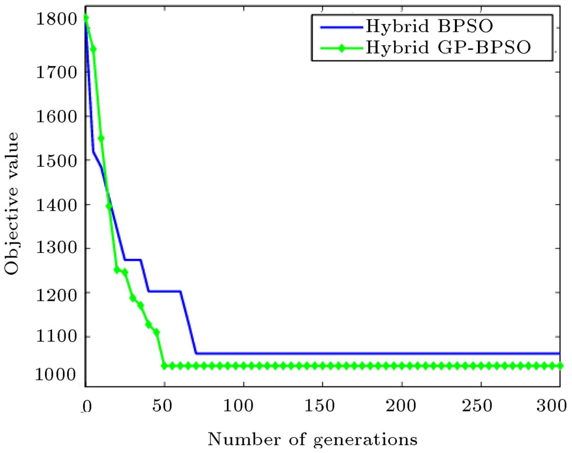

Figure 4. Performance comparison with $GEN = 300$, $\alpha = 0.15$, and $K = 1000$.

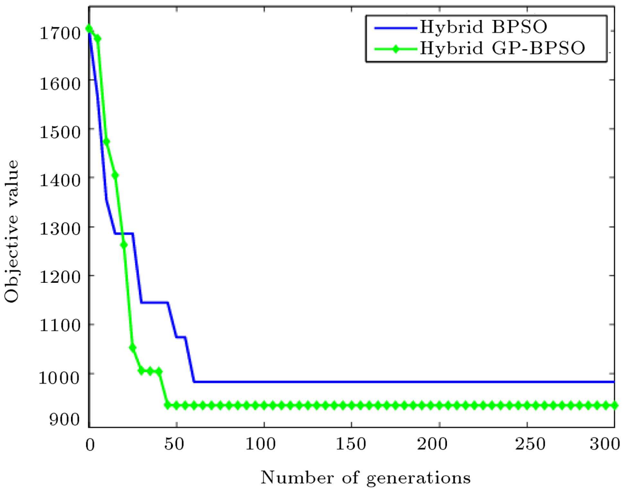

Figure 5. Performance comparison with $GEN = 300$, $\alpha = 0.20$, and $K = 1000$.

Then, DPM was incorporated into hybrid GP-BPSO method, which constituted a novel algorithm to simulate the approximate subproblem; DPM was utilized for computing the objective of the lower-level problem. By performing a series of simulated examples, we observed that the method proposed above was both feasible and effective.

References

1. Joseph, M., Cecil, P.R., and Edward, D.W., *Project Management with CPM, PERT, and Precedence Diagramming*, Van Nostrand Reinhold Pub. (1983).
2. Chen, Y.L., Rinks, D., and Tang, K. "Critical path in an activity network with time constraints", *Eur. J. Oper. Res.*, **100**(1), pp. 122-133 (1997).
3. Guerriero, F. and Talarico, L. "A solution approach to find the critical path in a time-constrained activity network", *Comput. Oper. Res.*, **37**(9), pp. 1557-1569 (2010).
4. Möhring, R.H. "Minimizing costs of resource requirements in project networks subject to a fixed completion time", *Oper. Res.*, **32**(1), pp. 89-120 (1984).
5. Mitchell, G. and Klastorin, T. "An effective methodology for the stochastic project compression problem", *IIE Trans.*, **39**(10), pp. 957-969 (2007).
6. Shen, S., Smith, J.C., and Ahmed, S. "Expectation and chance-constrained models and algorithms for insuring critical paths", *Manage. Sci.*, **56**(10), pp. 1794-1814 (2010).
7. Goh, J. and Hall, N.G. "Total cost control in project management via satisficing", *Manage. Sci.*, **59**(6), pp. 1354-1372 (2013).
8. Li, Z., Liu, Y., and Zang, W. "Stochastic insuring critical path problem with value-at-risk criterion", *Lecture Notes in Electrical Engineering*, **3**, pp. 3-10 (2013).
9. Li, Z., Liu, Y., and Yang, G. "A new probability model for insuring critical path problem with heuristic algorithm", *Neurocomputing*, **148**, pp. 129-135 (2015).
10. Li, H. and Womer, N.K. "Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming", *Eur. J. Oper. Res.*, **246**(1), pp. 20-33 (2015).
11. Zadeh, L.A. "Fuzzy sets", *Information and Control*, **8**(3), pp. 338-353 (1965).
12. Chen, S.P. and Hsueh, Y.J. "A simple approach to fuzzy critical path analysis in project networks", *Appl. Math. Model.*, **32**(7), pp. 1289-1297 (2008).
13. Zammori, F.A., Braglia, M., and Frosolini, M. "A fuzzy multi-criteria approach for critical path definition", *International Journal of Project Management*, **27**(3), pp. 278-291 (2009).
14. Amiri, M. and Golozari, F. "Application of fuzzy multi-attribute decision making in determining the critical path by using time, cost, risk, and quality criteria", *The International Journal of Advanced Manufacturing Technology*, **54**(1-4), pp. 393-401 (2011).
15. Zareei, A., Zaerpour, F., Bagherpour, M., Noora, A.A., and Vencheh, A.H. "A new approach for solving fuzzy critical path problem using analysis of events", *Expert Syst. Appl.*, **38**(1), pp. 87-93 (2011).
16. Li, Z. and Dai, X. "Optimization of insuring critical path problem with uncertain activity duration times", *Journal of Uncertain Systems*, **7**(1), pp. 72-80 (2013).
17. Kaur, P. and Kumar, A. "Linear programming approach for solving fuzzy critical path problems with fuzzy parameters", *Appl. Soft Comput.*, **21**, pp. 309-319 (2014).
18. Jayagowri, P. and Geetharamani, G. "Using metric distance ranking method to find intuitionistic fuzzy critical path", *J. Appl. Math.*, **2015**(20), pp. 1-12 (2015).
19. Pelikan, M., Stikova, H., and Vrana, I. "Detection of resource overload in conditions of project ambiguity", *IEEE Trans. Fuzzy Syst.* (2016). DOI: 10.1109/TFUZZ.2016.2584645
20. Sireesha, V. and Shankar, N.R. "A node-weighted rooted tree (NWRT) method to find project characteristics and critical path in a triangular fuzzy project network", *Comput. Appl. Math.*, pp. 1-15 (2017). DOI: 10.1007/s40314-017-0434-0
21. Liu, Y.K. and Liu, B. "Fuzzy random variables: A scalar expected value operator", *Fuzzy Optim. Decis. Ma.*, **2**(2), pp. 143-160 (2003).
22. Liu, Y.K., Liu, Z.Q., and Gao, J. "The modes of convergence in the approximation of fuzzy random optimization problems", *Soft Comput.*, **13**(2), pp. 117-125 (2009).
23. Qin, R. and Liu, Y.K. "A new data envelopment analysis model with fuzzy random inputs and outputs", *Journal of Applied Mathematics and Computing*, **33**(1-2), pp. 327-356 (2010).
24. Ma, Y. and Xu, J. "A novel multiple decision-maker model for resource-constrained project scheduling problems", *Can. J. of Civil Eng.*, **41**(6), pp. 500-511 (2014).
25. Yang, K. and Liu, Y. "Developing equilibrium optimization methods for hub location problems", *Soft Comput.*, **19**(8), pp. 2337-2353 (2015).
26. Gao, Y. and Qin, Z. "A chance constrained programming approach for uncertain p-hub center location problem", *Comput. Ind. Eng.*, **102**, pp. 10-20 (2016).
27. Meiyi, W., Xiang, L., and Lean, Y. "Time-dependent fuzzy random location-scheduling programming for hazardous materials transportation", *Transportation Research. Part C: Emerging Technologies*, **57**, pp. 146-165 (2015).
28. Kumar, R.S., Tiwari, M.K., and Goswami, A. "Two-echelon fuzzy stochastic supply chain for the manufacturer-buyer integrated production-inventory system", *J. Intell. Manuf.*, **27**(4), pp. 875-888 (2016).
29. Chen, Y., Gao, J., Yang, G., and Liu, Y. "Solving equilibrium standby redundancy optimization problem by hybrid PSO algorithm", *Soft Comput.*, **22**(17), pp. 5631-5645 (2018). DOI 10.1007/s00500-017-2552-4
30. Liu, Y.K. and Liu, B. "Fuzzy random programming with equilibrium chance constraints", *Inform. Sciences*, **170**(2), pp. 363-395 (2005).
31. Phelps, C., Royset, J.O., and Gong, Q. "Optimal control of uncertain systems using sample average

approximations", *SIAM J. Control Optim.*, **54**(1), pp. 1-29 (2016).

32. Kennedy, J. and Eberhart, R.C. "A discrete binary version of the particle swarm algorithm", In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997 IEEE International Conference*, **5** pp. 4104-4108, IEEE (Oct. 1997).

33. Lee, S., Soak, S., Oh, S., Pedrycz, W., and Jeon, M. "Modified binary particle swarm optimization", *Progress in Natural Science*, **18**(9), pp. 1161-1166 (2008).

34. Wang, S. and Watada, J. "A hybrid modified PSO approach to VaR-based facility location problems with variable capacity in fuzzy random uncertainty", *Inform. Sciences*, **192**, pp. 3-18 (2012).

35. Beheshti, Z., Shamsuddin, S.M., and Hasan, S. "Memetic binary particle swarm optimization for discrete optimization problems", *Inform. Sciences*, **299**, pp. 58-84 (2015).

36. Li, P., Xu, D., Zhou, Z., Lee, W.J., and Zhao, B. "Stochastic optimal operation of microgrid based on chaotic binary particle swarm optimization", *IEEE Transactions on Smart Grid*, **7**(1), pp. 66-73 (2016).

37. Liu, J., Mei, Y., and Li, X. "An analysis of the inertia weight parameter for binary particle swarm optimization", *IEEE Trans. Evolut. Comput.*, **20**(5), pp. 666-681 (2016).

38. Jiang, F., Xia, H., Tran, Q.A., Ha, Q.M., Tran, N.Q., and Hu, J. "A new binary hybrid particle swarm optimization with wavelet mutation", *Knowledge-Based Syst.*, **130**, pp. 90-101 (2017).

39. Liu, B. and Liu, Y.K. "Expected value of fuzzy variable and fuzzy expected value models", *IEEE Trans. Fuzzy Syst.*, **10**(4), pp. 445-450 (2002).

40. Liu, Y.K. and Gao, J. "The independence of fuzzy variables with applications to fuzzy random optimization", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **15**(supp02), pp. 1-20 (2007).

41. Shapiro, A. and Homem-de-Mello, T. "A simulation-based approach to two-stage stochastic programming with recourse", *Math. Program.*, **81**(3), pp. 301-325 (1998).

42. Luedtke, J. and Ahmed, S. "A sample approximation approach for optimization with probabilistic constraints", *SIAM J. Optimiz.*, **19**(2), pp. 674-699 (2008).

43. Bellman, R. "Dynamic programming and Lagrange multipliers", *P. Natl. A. Sci.*, **42**(10), pp. 767-769 (1956).

44. Huang, Y., Qu, L., and Tang, C. "Optimal coverage scheme based on QPSO in wireless sensor networks", *Journal of Networks*, **7**(9), pp. 1362-1368 (2012).

Biographies

Zhenhong Li is a PhD candidate in the College of Management and Economics at Tianjin University. She received her BS degree from the Department of Mathematics and Information Technology, Tangshan Normal College, in 2010 and her MS degree in Operations Research from Hebei University in 2013. Her research interests include stochastic optimization theory and its application in engineering problems. She has published a paper in Neurocomputing.

Bo Li is a Professor in the College of Management and Economics at Tianjin University. Her research interests include dual-channel competition and coordination in supply chains, game theory, and intelligent computation. She has published papers in Computers and Operations Research, International Journal of Production Economics, International Journal of Production Research, Annals of Operations Research, Journal of Cleaner Production, Applied Mathematical Modelling, International Journal of Information Systems and Supply Chain Management, etc.