Information Sciences 192 (2012) 3–18

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A hybrid modified PSO approach to VaR-based facility location problems with variable capacity in fuzzy random uncertainty

Shuming Wang, Junzo Watada *

Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135, Fukuoka, Japan

ARTICLE INFO

Article history:
Received 12 August 2009
Received in revised form 23 January 2010
Accepted 8 February 2010
Available online 14 February 2010

Keywords:
Facility location
Fuzzy random variable
Value-at-Risk
Particle swarm optimization
Mixed 0–1 integer fuzzy random programming
Variable capacity

ABSTRACT

This paper studies a facility location model with fuzzy random parameters and its swarm intelligence approach. A Value-at-Risk (VaR) based fuzzy random facility location model (VaR-FRFLM) is built in which both the costs and demands are assumed to be fuzzy random variables, and the capacity of each facility is unfixed but a decision variable assuming continuous values. Under this setting, the VaR-FRFLM is inherently a two-stage mixed 0–1 integer fuzzy random programming problem, to which analytical nonlinear programming methods are not applicable.

A hybrid modified particle swarm optimization (MPSO) approach is proposed to solve the VaR-FRFLM. In this hybrid mechanism, an approximation algorithm is utilized to compute the fuzzy random VaR objective, a continuous Nbest–Gbest-based PSO and a genotype–phenotype-based binary PSO vehicles are designed to deal with the continuous capacity decisions and the binary location decisions, respectively, and two mutation operators are incorporated into the PSO to further decrease the possibility of becoming trapped in the local optima. A numerical experiment illustrates the application of the proposed hybrid MPSO algorithm and lays out its robustness to the system parameter settings. The comparison shows that the hybrid MPSO exhibits better performance than that when hybrid regular continuous-binary PSO and genetic algorithm (GA) are used to solve the VaR-FRFLM.

1. Introduction

Facility location selection is a category of optimization problems that aim to maximize the return or minimize the costs via determining the locations of facilities to open from a set of potential sites. Various kinds of facility location problems with uncertainty have been investigated in the literature. The first category is the stochastic facility location problems which deal with the cases when the uncertain parameters, like customers' demands and operating costs of plants, are characterized by random variables. For instance, Logendran and Terrell [22] developed a stochastic uncapacitated transportation plant location-allocation model with the objective of maximizing the expected profits, and they proposed some heuristics to solve the problem. Louveaux and Peeters [23] discussed a dual-based procedure for a stochastic facility location problem with recourse. Laporte et al. [14] formulated a class of capacitated facility location problems with random demands by using stochastic integer linear programming, and proposed a branch and cut solution approach. Schutz et al. [29] considered a stochastic facility location problem with general long-run costs and convex short-run costs, and solved the problem through

* Corresponding author.
E-mail addresses: smwangips@gmail.com (S. Wang), junzow@osb.att.ne.jp (J. Watada).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.02.014

a Lagrange relaxation based method. For more details on stochastic facility location problems, one may refer to Berman and Drezner [1], Carrizosa et al. [3], and Zeng and Ward [40].

Another category of facility location problems with uncertain parameters was developed based on fuzzy set theory [25,26,39] and possibility theory [5,18,37], which deal with imprecise or vague parameters in location problems. For instance, Bhattacharya et al. [2] considered facilities located under multiple fuzzy criteria and proposed a fuzzy goal programming to cope with the problem. Ishii et al. [7] developed a location model by considering the satisfaction degree expressed with respect to the distance from the facility to each customer. Lin and Li [16] discussed a fuzzy decision support system for selecting the appropriate facility site among countries of multinational enterprises. Introducing the fuzzy Value-at-Risk (VaR) into the facility location problems, Wang et al. [32] built a VaR-based two-stage fuzzy facility location model and discussed its metaheuristic approaches. Assuming that demands of customers are represented as fuzzy variables, Wen and Iwamura [34] presented a continuous α-cost facility location model employing the Hurwicz criterion, and Zhou and Liu [41] presented three types of continuous capacitated location-allocation problems with different decision criteria.

In real-world applications, stochastic variability (randomness) and vagueness or fuzziness may coexist in one facility location problem. On the one hand, due to the subjective judgement and imprecise human knowledge and perception in capturing statistic data, the parameters of the real location problems may embrace randomness and fuzziness at the same time. On the other hand, sometimes the historical data available for the parameter-distributions in location problems are insufficient, therefore, the expert knowledge (fuzzy information) should be incorporated into the available statistical data. In both cases mentioned above, there is a genuine need to deal with a hybrid uncertainty of randomness and fuzziness in location problems. Making use of the expected value operator of a fuzzy random variable [11,12,19,24] as the objective, Wang et al. [33] modeled a recourse-based facility location problem with fuzzy random uncertainty and discussed its binary particle swarm optimization (BPSO) approach. Developing the work [34] in a hybrid uncertain environment, Wen and Iwamura [35] built an (α,β)-cost minimization model for random fuzzy facility location problems under the Hurwicz criterion, and designed a genetic algorithm (GA) for continuous decision variables in the location model.

In both [33,35], the capacity of each facility was assumed to be fixed. However, in more realistic situations, it should be variable and modeled as a decision to be made. In this paper, we assume the capacity to be a decision variable and build a fuzzy random facility location model with an objective of fuzzy random VaR (see [31]). In contrast with the model in [33] whose decisions are all binary variables, and the model in [35] where all the decisions are continuous variables, the fuzzy random location model built in this paper contains mixed decisions. More precisely, the capacities are the continuous decisions while the location decisions are binary ones. The proposed location model of this nature is essentially a two-stage mixed 0–1 integer fuzzy random programming problem. As a consequence, we design a hybrid approach to this new model that comprises an approximation algorithm to fuzzy random VaR and a mechanism of modified continuous-binary PSO. The proposed hybrid approach is compared with other hybrid approaches, such as hybrid GA and regular PSO.

The material is organized as follows. Section 2 recalls some preliminaries on fuzzy random variables and fuzzy random VaR. In Section 3, we formulate the problem and highlight its difficulties. Section 4 discusses the solution approach: the hybrid MPSO algorithm. The numerical experiments and comparisons are provided in Section 5. Section 6 covers conclusions.

2. Preliminaries

In this section, we briefly review some essentials of fuzzy random variables, as for more detailed discussion on fuzzy random variables, one may refer to Kruse and Meyer [11], Kwakernaak [12], López-Diaz and Gil [24], Liu [17], and Zadeh [38].

Let the triplet $(\Gamma,\mathcal{P}(\Gamma),\mathrm{Pos})$ be a possibility space, where $\mathcal{P}(\Gamma)$ is the power set of Γ, X be a fuzzy variable defined on $(\Gamma,\mathcal{P}(\Gamma),\mathrm{Pos})$ whose membership function is μ_X. The possibility and credibility of an event $X\leqslant r$ are expressed as follows:

$$\begin{aligned}&\mathrm{Pos}\{X\leqslant r\}=\sup_{t\leqslant r}\mu_X(t),\quad \text{and}\\&\mathrm{Cr}\{X\leqslant r\}=\frac{1}{2}\left(\sup_{t\leqslant r}\mu_X(t)+1-\sup_{t>r}\mu_X(t)\right),\end{aligned} \tag{1}$$

where r is any real number.

Suppose that $(\Omega,\mathcal{A},\mathrm{Pr})$ is a probability space, $\mathcal{F}_v$ is a collection of fuzzy variables defined on possibility space $(\Gamma,\mathcal{P}(\Gamma),\mathrm{Pos})$. A fuzzy random variable is defined as a map $\xi:\Omega\to\mathcal{F}_v$ such that $\mathrm{Pos}\{\xi(\omega)\in B\}$ is a measurable function of ω for any Borel subset B of $\Re$ (see [19]).

Example 1. Let Y be a random variable defined on probability space $(\Omega,\mathcal{A},\mathrm{Pr})$. If we define that for every $\omega\in\Omega,\xi(\omega)=(Y(\omega),Y(\omega)+2,Y(\omega)+6)$ which is a triangular fuzzy variable defined on some possibility space $(\Gamma,\mathcal{P}(\Gamma),\mathrm{Pos})$. Then, ξ is a (triangular) fuzzy random variable.

To measure an event $\xi\in B$ induced by a fuzzy random variable ξ, where B a Borel subset of $\Re$, the mean chance measure (see [20]) is given as

$$\mathrm{Ch}\{\xi\in B\}=\int_\Omega \mathrm{Cr}\{\xi(\omega)\in B\}\mathrm{Pr}(\,\mathrm{d}\omega). \tag{2}$$

The Value-at-Risk (VaR) of an investment is the likelihood of the greatest loss with some confidence level (see [6]). In the fuzzy random environment, let $\mathcal{L}$ be the loss variable with fuzzy random parameters of some investment. The fuzzy random Value-at-Risk of the investment with confidence $1-\beta$ is expressed in the following form (see [31]):

Definition 1. We let $\mathcal{L}$ be the loss variable with fuzzy random parameters of some investment. The fuzzy random Value-at-Risk of the investment with confidence $1-\beta$ is expressed in the following form:

$$\mathrm{VaR}_{1-\beta} = \sup\{\lambda \in \Re | \mathrm{Ch}\{\mathcal{L} \geqslant \lambda\} \geqslant \beta\}, \tag{3}$$

where $\beta \in (0,1)$, and Ch is the mean chance measure in (2).

Remark 1. If the fuzzy random loss variable $\mathcal{L}$ in Definition 1 degenerates to a random loss variable, then we have

$$\begin{aligned}\mathrm{VaR}_{1-\beta} &= \sup\{\lambda \in \Re | \mathrm{Ch}\{\mathcal{L} \geqslant \lambda\} \geqslant \beta\} = \sup\left\{\lambda \in \Re | \int_{\Omega} \mathrm{Cr}\{\mathcal{L}(\omega) \geqslant \lambda\}\mathrm{Pr}(\mathrm{d}\omega) \geqslant \beta\right\} \\ &= \sup\left\{\lambda \in \Re | \int_{\Omega} I_{\{\mathcal{L}(\omega)\geqslant\lambda\}}\mathrm{Pr}(\mathrm{d}\omega) \geqslant \beta\right\} = \sup\{\lambda \in \Re | \mathrm{Pr}\{\mathcal{L} \geqslant \lambda\} \geqslant \beta\}\end{aligned}$$

where $I_{\{\cdot\}}$ is the indicator function of event $\{\cdot\}$. Hence, the Value-at-Risk in (3) becomes the classic stochastic Value-at-Risk (see [6]).

Remark 2. If the fuzzy random loss variable $\mathcal{L}$ in Definition 1 reduces to a fuzzy loss variable, then $\mathcal{L}(\omega) \equiv \mathcal{L}$ for any $\omega \in \Omega$. Thus, the Value-at-Risk in (3) becomes

$$\mathrm{VaR}_{1-\beta} = \sup\left\{\lambda \in \Re | \int_{\Omega} \mathrm{Cr}\{\mathcal{L} \geqslant \lambda\}\mathrm{Pr}(\mathrm{d}\omega) \geqslant \beta\right\} = \sup\{\lambda \in \Re | \mathrm{Cr}\{\mathcal{L} \geqslant \lambda\} \geqslant \beta\},$$

which is just the fuzzy Value-at-Risk (see [32]).

Example 2. Suppose the loss $\mathcal{L}$ of an investment is a triangular fuzzy random variable $(X-100, X, X+100)$ (\$), where X is a discrete random variable taking on values $X_1 = 50$ with probability 0.8 and $X_2 = 100$ with probability 0.2, respectively. Let us calculate the VaR at a confidence of 0.9.

From the assumptions, we know the loss $\mathcal{L}$ takes on the fuzzy values as follows: $\mathcal{L}(X_1) = (-50, 50, 150)$ with probability 0.8, and $\mathcal{L}(X_2) = (0, 100, 200)$ with probability 0.2. From (1), we can compute

$$\mathrm{Cr}\{\mathcal{L}(X_1) \geqslant r\} = \begin{cases} 1, & r \leqslant -50, \\ (150-r)/200, & -50 < r \leqslant 150, \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mathrm{Cr}\{\mathcal{L}(X_2) \geqslant r\} = \begin{cases} 1, & r \leqslant 0, \\ (200-r)/200, & 0 < r \leqslant 200, \\ 0, & \text{otherwise.} \end{cases}$$

From the definition, we obtain

$$\begin{aligned}\mathrm{Ch}\{\mathcal{L} \geqslant r\} &= 0.8 \times \mathrm{Cr}\{\mathcal{L}(X_1) \geqslant r\} + 0.2 \times \mathrm{Cr}\{\mathcal{L}(X_2) \geqslant r\} \\ &= \begin{cases} 1, & r \leqslant -50, \\ (200-r)/250, & -50 < r \leqslant 0, \\ (160-r)/200, & 0 < r \leqslant 150, \\ (200-r)/1000, & 150 < r \leqslant 200, \\ 0, & \text{otherwise.} \end{cases}\end{aligned}$$

Therefore, by (3), the Value-at-Risk at the confidence level of 0.9 is

$$\mathrm{VaR}_{0.9} = \sup\{r | (160-r)/200 \geqslant 0.1\} = 140\$.$$

3. VaR-based fuzzy random facility location model

In this section, we formulate a VaR-based fuzzy random facility location model (VaR-FRFLM) with variable capacity. We introduce the following notation for this location model:*Indices and constants*

i index of facilities, $1 \leqslant i \leqslant n$

j index of clients, $1 \leqslant j \leqslant m$

r_j unit price charged to client j
c_i fixed cost for opening and operating facility i
W_i maximum capacity of each facility i
t_{ij} unit transportation cost from i to j
$1-\beta$ confidence level of the Value-at-Risk

Fuzzy random parameters
D_j fuzzy random demand of client j
V_i fuzzy random unit variable operating cost of facility i
ξ fuzzy random demand-cost vector $\xi = (D_1, \ldots, D_m, V_1, \ldots, V_n)$

Decision variables
x_i location decision which is a binary variable
$\boldsymbol{x}$ location decision vector which is $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
s_i capacity decision of facility i
$\boldsymbol{s}$ capacity decision vector which is $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$
$y_{ij}^{(\omega,\gamma)}$ quantity supplied to client j from facility i at scenario (ω, γ)

As usual, it is assumed that each customer's demand cannot be over served, but it is possible that not all demand is served. Furthermore, the total supply from one facility to all clients cannot exceed the capacity of the facility. Lastly, we assume that fuzzy random demand-cost vector $\xi = (D_1, \ldots, D_m, V_1, \ldots, V_n)$ is defined from a probability space $(\Omega, \mathcal{A}, \Pr)$ to a collection of fuzzy vectors on possibility space $(\Gamma, \mathcal{P}(\Gamma), \mathrm{Pos})$.

Making use of the fuzzy random VaR in (3), a VaR-FRFLM at confidence level $1-\beta$ can be built as follows under the above notation and assumptions.

Model

$$\left.\begin{array}{ll} \min & \mathrm{VaR}_{1-\beta}(\boldsymbol{x}, \boldsymbol{s}) \\ \text{subject to} & x_i \in \{0,1\}, \quad i = 1, 2, \ldots, n, \\ & 0 \leqslant s_i \leqslant W_i x_i, \quad i = 1, 2, \ldots, n, \end{array}\right\} \tag{4}$$

where

$$\mathrm{VaR}_{1-\beta}(\boldsymbol{x}, \boldsymbol{s}) = \sup\left\{\lambda \,\middle|\, \mathrm{Ch}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\boldsymbol{x}, \boldsymbol{s}, \xi) \geqslant \lambda\right\} \geqslant \beta\right\} \tag{5}$$

and the second-stage problem for each scenario (ω, γ) is

$$\left.\begin{array}{lll} \mathcal{R}(\boldsymbol{x}, \boldsymbol{s}, \xi(\omega,\gamma)) = \max & & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m}(r_j - V_i(\omega,\gamma) - t_{ij}) y_{ij}^{(\omega,\gamma)} \\ & \text{subject to} & \displaystyle\sum_{i=1}^{n} y_{ij}^{(\omega,\gamma)} \leqslant D_j(\omega,\gamma), \quad j = 1, 2, \ldots, m, \\ & & \displaystyle\sum_{j=1}^{m} y_{ij}^{(\omega,\gamma)} \leqslant s_i x_i, \quad i = 1, 2, \ldots, n, \\ & & y_{ij}^{(\omega,\gamma)} \geqslant 0, \quad i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m. \end{array}\right\} \tag{6}$$

The objective of this VaR-FRFLM (4)–(6) is to minimize the VaR of the investment by determining the optimal locations as well as the capacities of the new facilities to open. Here $\sum_{i=1}^{n} c_i x_i$ represents the fixed cost at each location decision $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, and $\mathcal{R}(\boldsymbol{x}, \boldsymbol{s}, \xi)$ is the variable return based on the fuzzy random parameter ξ. Herewith, the $\mathrm{VaR}_{1-\beta}(\boldsymbol{x}, \boldsymbol{s})$ in (5) represents the largest loss at the confidence $1-\beta$.

In the VaR-FRFLM, the location-capacity decision $(\boldsymbol{x}, \boldsymbol{s})$ is called the first stage decision in the theory of two-stage fuzzy random programming with VaR criteria (see [31]), which should be made before the realizations $D_j(\omega, \gamma)$ and $V_i(\omega, \gamma)$ of the fuzzy random demand D_j and cost V_i are observed, respectively, where the scenario $(\omega, \gamma) \in \Omega \times \Gamma$. Furthermore, we note that the objective function is

$$\begin{aligned} \mathrm{VaR}_{1-\beta}(\boldsymbol{x}, \boldsymbol{s}) &= \sup\left\{\lambda \,\middle|\, \mathrm{Ch}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\boldsymbol{x}, \boldsymbol{s}, \xi) \geqslant \lambda\right\} \geqslant \beta\right\} \\ &= \sup\left\{\lambda \,\middle|\, \int_{\Omega} \mathrm{Cr}\left\{\gamma \in \Gamma \,\middle|\, \sum_{i=1}^{n} c_i x_i - \mathcal{R}(\boldsymbol{x}, \boldsymbol{s}, \xi(\omega,\gamma)) \geqslant \lambda\right\} \Pr(\mathrm{d}\omega) \geqslant \beta\right\}, \end{aligned} \tag{7}$$

for each first stage decision $(\boldsymbol{x},\boldsymbol{s})$, hence, in order to determine the value of objective $\mathrm{VaR}_{1-\beta}(\boldsymbol{x},\boldsymbol{s})$ we have to solve N second stage problems (6), where N is the number of all the scenarios $(\omega,\gamma)\in\Omega\times\Gamma$. Given $(\boldsymbol{x},\boldsymbol{s})$, for each scenario (ω,γ), the quality distribution pattern $(y_{ij}^{(\omega,\gamma)})_{n\times m}$ is determined by solving the second stage problem (6) at scenario (ω,γ). Here, the $y_{ij}^{(\omega,\gamma)}$ for $i=1,2,\ldots,n;j=1,2,\ldots,m$ are referred to as the second stage decisions (see [31]). From the model (4)–(6), we can see the second stage decision $y_{ij}^{(\omega,\gamma)}$ is up to the scenario $(\omega,\gamma)\in\Omega\times\Gamma$, it does not serve as the decision to all the scenarios but is determined for further calculation of $\mathrm{VaR}_{1-\beta}(\boldsymbol{x},\boldsymbol{s})$. So the real decision in VaR-FRFLM is the first stage decision $(\boldsymbol{x},\boldsymbol{s})$.

It is easy to see that the VaR-FRFLM is a task of two-stage mixed 0–1 integer fuzzy random programming. In general, the fuzzy random parameter V_i and D_j for $i=1,2,\ldots,n;j=1,2,\ldots,m$ are continuous fuzzy random variables which have an infinite number of realizations. Hence, from (7) we can see that it requires to solve an infinite number of second-stage problems (6) so as to determine the objective value $\mathrm{VaR}_{1-\beta}(\boldsymbol{x},\boldsymbol{s})$. It cannot be calculated analytically. As a consequence, the VaR-FRFLM (4)–(6) within this nature cannot be solved analytically, we will design a hybrid metaheuristic approach to this two-stage mixed 0-1 fuzzy random programming problem.

4. Hybrid MPSO approach

Particle Swarm Optimization (PSO) is a biologically inspired evolutionary computation algorithm, which was originally developed by Kennedy and Eberhart [8]. This technique uses collaboration among a population of simple search agents (called particles) to find optima in some search space, and has been shown to be effective in optimizing difficult multidimensional problems in a variety of fields (see [10,13,27,28,31]). Since the standard PSO is a real-coded algorithm, in order to cope with binary-coded optimization problems, a binary PSO (BPSO) was also introduced (see [9]). BPSO has been found to be robust in solving optimization problems featuring binary decision variables (see [15,30,32,33]).

Recall that the proposed VaR-FRFLM (4)–(6) is a two-stage mixed 0–1 integer fuzzy random programming problems. We design a hybrid mechanism, which integrates the continuous PSO, BPSO, and approximation algorithm to fuzzy random VaR, to solve the model. Several modifications are made so as to enhance the performance of the hybrid approach:

(i) We employ a phenotype-genotype mechanism (see [15,32]) in the BPSO to further enhance the searching capability of the binary particles.
(ii) Most PSO algorithms employ the type of Pbest–Gbest-based update formula (the 'Pbest' and 'Gbest' denote personal best and global best particles, respectively). However, in population-based optimization mechanism, an individual should be influenced not only by the personal best position of itself and the global best position of the population, but also by that of its neighbors (actually, such neighborhood-based individual development mode is obvious from the social point of view). Thus, to further improve the global search in the optimization, it is desirable to consider the individual's neighborhood which is better than singly considering the individual itself. From this point of view, we introduce a Nbest–Gbest-based update rule (the 'Nbest' denotes the neighborhood-best particles) by adjusting the velocity in the directions of the personal best particles in the neighborhood and the global best particle.
(iii) Two mutation operators are applied to the binary location particles and capacity particles, respectively, to decrease the probability of its getting trapped in a local optimum so as to further enhance the search ability of the hybrid algorithm.

The proposed hybrid algorithm is referred to as a hybrid modified PSO (MPSO) algorithm, which is elaborated as follows.

4.1. Approximation to fuzzy random VaR

An approximation algorithm for fuzzy random VaR has been proposed in the VaR-based two-stage fuzzy stochastic programming (see [31]), and the convergence of the approximation algorithm is also proved in [31]. In this paper, we employ the approximation algorithm to estimate the objective value

$$\mathrm{VaR}_{1-\beta}(\boldsymbol{x},\boldsymbol{s})=\sup\left\{\lambda\middle|\ \mathrm{Ch}\left\{\sum_{i=1}^{n}c_ix_i-\mathcal{R}(\boldsymbol{x},\boldsymbol{s},\xi)\geqslant\lambda\right\}\geqslant\beta\right\}\tag{8}$$

for each $(\boldsymbol{x},\boldsymbol{s})$ in our VaR-FRFLM (4)–(6). The detailed approximation algorithm (Algorithms 2 and 3) to fuzzy random VaR is given in Appendix A.

4.2. Solution representation

A real number vector $(\boldsymbol{x},\boldsymbol{s})\triangleq(\langle x_1,s_1\rangle,\langle x_2,s_2\rangle,\ldots,\langle x_n,s_n\rangle)$ is used as a particle pair to represent a solution (location-capacity) of the two-stage VaR-FRFLM (4)–(6), where $x_i\in\{0,1\},0\leqslant s_i\leqslant W_ix_i,i=1,2,\ldots,n$.

4.3. Initialization

First of all, we randomly generate the initial binary phenotype location particle $\boldsymbol{x}_p=(x_{p,1},x_{p,2},\ldots,x_{p,n})$ as follows:

$$\begin{array}{l}\texttt{for}(i = 1; i <= n; i++) \\ \quad \texttt{if}(\text{rand}() > 0.5)\ \texttt{then}\ x_{p,i} = 1;\ \texttt{else}\ x_{p,i} = 0;\end{array} \tag{9}$$

where rand() is a random number coming from the uniform distribution over the interval [0,1], and initialize the genotype location particle $\mathrm{x}_g = \mathrm{x}_p$. Then, we generate a capacity particle $\mathrm{s} = (s_1, s_2, \ldots, s_n)$ by the following method:

$$\begin{array}{l}\texttt{for}(i = 1; i <= n; i++) \\ \quad \texttt{if}(x_{p,i} = 1)\ \texttt{then}\ s_i = \text{rand}(0, W_i);\ \texttt{else}\ s_i = 0;\end{array} \tag{10}$$

where rand(a,b) is a uniformly distributed random number over the interval [a,b]. Repeat the above process P_{size} times, we get P_{size} initial binary phenotype and genotype location particles $\mathrm{x}_{p,1}, \mathrm{x}_{p,2}, \ldots, \mathrm{x}_{p,P_{size}}; \mathrm{x}_{g,1}, \mathrm{x}_{g,2}, \ldots, \mathrm{x}_{g,P_{size}}$, and P_{size} capacity particles $\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_{P_{size}}$, respectively.

4.4. Evaluation by approximation algorithm to VaR

Denote **Fit**$(\cdot)$ the fitness function, and let the fitness of each decision (x, s) be the minus of the Value-at-Risk, i.e.,

$$\mathbf{Fit}(\mathrm{x}, \mathrm{s}) = -\mathrm{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}).$$

Therefore, the particles of smaller objective values are evaluated with higher fitness. For each (x, s), the fitness value **Fit**(x, s) is calculated by the approximation algorithm mentioned in Sub Section 4.1.

4.5. Update process

4.5.1. Update of genotype-location and capacity particles

In the update process, we first need to determine the global best particle pair $(\mathrm{x}_{Gbest}, \mathrm{s}_{Gbest})$ (with the highest fitness), where the x_{Gbest} is the best phenotype location particle so far; and for each $(\mathrm{x}_{p,k}, \mathrm{s}_k)$, find the $(\mathrm{x}_{Pbest,k}, \mathrm{s}_{Pbest,k})$ with the highest previous fitness, where $k = 1, 2, \ldots, P_{size}$. Then, for each k, we determine the velocity vector pair $(\mathrm{v}_{\mathrm{x},k}, \mathrm{v}_{\mathrm{s},k})$ through the following *Nbest–Gbest-based* update formula (see Fig. 1):

$$\mathrm{v}_{\mathrm{x},k} = \mathcal{W} * \mathrm{v}_{\mathrm{x},k} + c_1 * d_N(\mathrm{x}_{p,k}) + c_2 * \text{rand}() * (\mathrm{x}_{Gbest} - \mathrm{x}_{p,k}), \tag{11}$$

$$\mathrm{v}_{\mathrm{s},k} = \mathcal{W} * \mathrm{v}_{\mathrm{s},k} + c_1 * d_N(\mathrm{s}_k) + c_2 * \text{rand}() * (\mathrm{s}_{Gbest} - \mathrm{s}_k). \tag{12}$$

In the above formula, $d_N(\mathrm{x}_{p,k}), k = 1, 2, \ldots, P_{size}$ are the average distances from $\mathrm{x}_{p,k}$ to the best positions in its neighborhood, which are defined as

$$d_N(\mathrm{x}_{p,1}) = \sum_{j=1}^{2} \text{rand}() * \left(\frac{\mathrm{x}_{Pbest,j} - \mathrm{x}_{p,1}}{2}\right), \tag{13}$$

$$d_N(\mathrm{x}_{p,k}) = \sum_{j=k-1}^{k+1} \text{rand}() * \left(\frac{\mathrm{x}_{Pbest,j} - \mathrm{x}_{p,k}}{3}\right), \quad k = 2, 3, \ldots, P_{size} - 1, \tag{14}$$

$$d_N\left(\mathrm{x}_{p,P_{size}}\right) = \sum_{j=P_{size}-1}^{P_{size}} \text{rand}() * \left(\frac{\mathrm{x}_{Pbest,j} - \mathrm{x}_{p,P_{size}}}{2}\right), \tag{15}$$

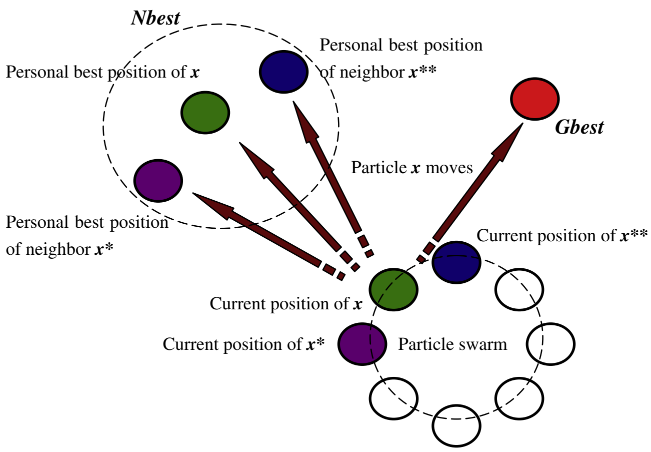

Fig. 1. Nbest–Gbest-based update process.

respectively, and the $d_N(\mathrm{s}_k), k = 1, 2, \ldots, P_{size}$ can be expressed similarly. Here, c_1 and c_2 are learning rates, to well adjust the convergence of the particles, we employ the time-varying learning rates (see [28]) as follows:

$$c_1 = 2\frac{G_{max} - G_n}{G_{max}} + 1, \tag{16}$$

$$c_2 = 2 * \frac{G_n}{G_{max}} + 1, \tag{17}$$

where G_{max} and G_n are the indexes of the maximum and current generations, respectively. $\mathcal{W}$ is the inertia weight which is set by the following expression [4]:

$$\mathcal{W} = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|},$$

where $\phi = c_1 + c_2$.

Next, each genotype location particle $\mathrm{x}_{g,k}$ and capacity particle s_k are updated by the following operations

$$\mathrm{x}_{g,k} = \mathrm{x}_{g,k} + \mathrm{v}_{\mathrm{x},k}, \tag{18}$$

$$\mathrm{s}_k = \mathrm{s}_k + \mathrm{v}_{\mathrm{s},k}, \tag{19}$$

respectively.

4.5.2. Update of phenotype-location particles and re-update of capacity particles

All the phenotype location particles $\mathrm{x}_{p,k}, k = 1, 2, \ldots, P_{size}$ are updated according to the following rule [15]:

$$\begin{array}{l} \mathtt{for}(i = 1; i <= n; i++) \\ \quad \mathtt{if}(\mathrm{rand}() < S(x_{g,ki}))\ \mathtt{then}\ x_{p,ki} = 1;\ \mathtt{else}\ x_{p,ki} = 0; \end{array} \tag{20}$$

where $x_{g,ki}$ and $x_{p,ki}$ are the components of the vectors $\mathrm{x}_{g,k}$ and $\mathrm{x}_{p,k}$, respectively, and $S(\cdot)$ is a sigmoid function with $S(x) = 1/1 + e^{-x}$. Furthermore, we re-update the capacity particles s_k with the following constraint:

$$\begin{array}{l} \mathtt{for}(i = 1; i <= n; i++) \\ \{ \\ \quad \mathtt{if}(x_{p,ki} = 0)\ \mathtt{then}\ s_{ki} = 0; \\ \ \ \mathtt{else} \\ \quad if(s_{ki} = 0)\ \mathtt{then}\ s_{ki} = \mathrm{rand}\ (0, W_i); \\ \} \end{array} \tag{21}$$

where s_{ki} is a component of capacity particle s_k, for $k = 1, 2, \ldots, P_{size}$.

Making use of formulas (11)–(21), we yield a new generation of phenotype-location and capacity particle pairs $(\mathrm{x}'_{p,1}, \mathrm{s}'_1), (\mathrm{x}'_{p,2}, \mathrm{s}'_2), \ldots, (\mathrm{x}'_{p,P_{size}}, \mathrm{s}'_{P_{size}})$.

4.5.3. Mutation

We predetermine two parameters $P_{m,L}, P_{m,C} \in (0, 1)$ representing the probability of mutation for the location and capacity particles, respectively. The following mutation operation is applied to all velocity vectors of location particles after the update process (18) of the genotype location particles:

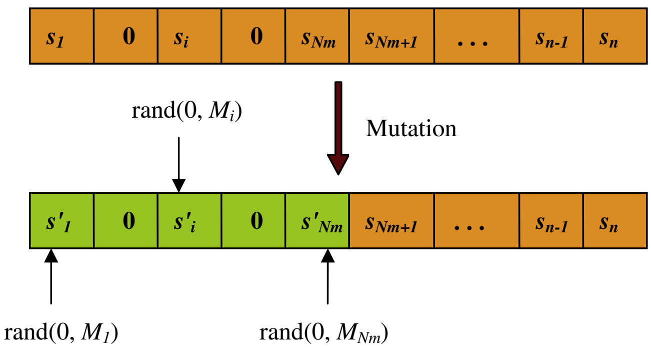

Fig. 2. Mutation of capacity particles.

$$\begin{array}{l} \texttt{for}(k=1; k <= P_{size}; k++) \\ \quad \texttt{if}(\text{rand}() < P_{m,L}) \texttt{ then } \mathrm{v}_{x,k} = -\mathrm{v}_{x,k}; \end{array} \tag{22}$$

On the other hand, the mutation of capacity particles is implemented following the update operation (21). For each capacity particle $\mathrm{s}_k = (s_{k1}, s_{k2}, \ldots, s_{kn}), k = 1, 2, \ldots, P_{size}$, if $\text{rand}() < P_{m,C}$, then we generate a number N_m between $1\&n$, and mutate the capacity particle as follows

$$\begin{array}{l} \texttt{for}(i=1; i <= N_m; i++) \\ \quad \texttt{if}(s_{ki} > 0) \texttt{ then } s_{ki} = \text{rand}(0, W_i); \end{array} \tag{23}$$

The mutation (23) can be shown in Fig. 2.

4.6. Hybrid algorithm procedure

The hybrid MPSO algorithm to VaR-FRFLM (4)–(6) can be summarized as follows.

Algorithm 1 (*Hybrid MPSO algorithm*).

Step 1. Initialize a population of phenotype-genotype location particles $\mathrm{x}_{p,k}, \mathrm{x}_{g,k}$, and capacity particles s_k, for $k = 1, 2, \ldots, P_{size}$, by using (9) and (10).
Step 2. Calculate the fitness $\textbf{Fit}(\mathrm{x}_p, \mathrm{s})$ for all particles through the approximation algorithm to VaR (Algorithms 2, 3 in Appendix A), and evaluate each particle pair according to the fitness;
Step 3. Determine the $d_N(\mathrm{x}_p)$ and $d_N(\mathrm{s})$ for each phenotype location particle x_p and capacity particle s, and find the global best particles x_{Gbest} and s_{Gbest} for the population;
Step 4. Update all the genotype location and capacity particles by formulas (11)–(19);
Step 5. Run mutation operator (22) to each location velocity with probability $P_{m,L}$.
Step 6. Update each phenotype location particle by (20), and re-update each capacity particle with (21).
Step 7. Run mutation operator (23) to each capacity particle with probability $P_{m,C}$.
Step 8. Repeat Steps 2–7 for a given number of generations;
Step 9. Return the particle pair $(\mathrm{x}_{Gbest}, \mathrm{s}_{Gbest})$ as the optimal solution to the VaR-FRFLM (4)–(6), and $\text{VaR}_{1-\beta}(\mathrm{x}_{Gbest}, \mathrm{s}_{Gbest}) = -\textbf{Fit}(\mathrm{x}_{Gbest}, \mathrm{s}_{Gbest})$ as the corresponding optimal value.

5. Numerical experiments and comparisons

We consider a firm which plans to open new facilities in 10 potential sites, the capacity limits W_i, fixed costs c_i and fuzzy random operating costs V_i of the sites $i, i = 1, 2, \ldots, 10$ are given in Table 1. We suppose that there are five customers whose fuzzy random demands $D_j, j = 1, 2, \ldots, 5$ are given in Table 2, where $\mathcal{U}(a, b)$ represents a random variable with uniform distribution on $[a, b]$, also the unit price r_j charged to each customer is also listed there. In addition, the unit transportation costs $t_{ij}, i = 1, 2, \ldots, 10; j = 1, 2, \ldots, 5$ are given by a matrix T as follows:

$$T = (t_{ij})_{5\times 10} = \left(\begin{array}{c|cccccccccc} & i=1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline j=1 & 16 & 21 & 19 & 18 & 14 & 18 & 16 & 20 & 18 & 20 \\ 2 & 17 & 15 & 17 & 14 & 18 & 16 & 17 & 18 & 17 & 14 \\ 3 & 24 & 20 & 25 & 22 & 23 & 22 & 24 & 22 & 20 & 22 \\ 4 & 19 & 22 & 18 & 15 & 21 & 17 & 22 & 21 & 22 & 16 \\ 5 & 13 & 10 & 16 & 13 & 14 & 11 & 13 & 15 & 14 & 13 \end{array}\right)$$

Within the above settings, we can formulate the location problem by using two-stage VaR-FRFLM as follows:

$$\left.\begin{array}{ll} \min & \text{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}) \\ \text{subject to} & x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, 10, \\ & 0 \leqslant s_i \leqslant M_i x_i, \quad i = 1, 2, \ldots, 10, \end{array}\right\} \tag{24}$$

where

$$\text{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}) = \sup\left\{\lambda \middle| \ \text{Ch}\left\{\sum_{i=1}^{10} c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \xi) \geqslant \lambda\right\} \geqslant \beta\right\} \tag{25}$$

Table 1
Capacity limits, fixed and variable costs.

Facility site i	Capacity limit W_i	Fixed cost c_i	Variable cost V_i	Parameter Y_i
1	250	8	$(7+Y_1, 9+Y_1, 10+Y_1)$	$\mathcal{U}(1,2)$
2	220	15	$(6+Y_2, 8+Y_2, 10+Y_2)$	$\mathcal{U}(2,3)$
3	300	16	$(8+Y_3, 10+Y_3, 11+Y_3)$	$\mathcal{U}(1,2)$
4	290	12	$(12+Y_4, 13+Y_4, 15+Y_4)$	$\mathcal{U}(0,1)$
5	260	6	$(13+Y_5, 15+Y_5, 16+Y_5)$	$\mathcal{U}(1,2)$
6	250	12	$(8+Y_6, 9+Y_6, 10+Y_6)$	$\mathcal{U}(0,2)$
7	320	17	$(6+Y_7, 7+Y_7, 8+Y_7)$	$\mathcal{U}(2,4)$
8	330	8	$(8+Y_8, 10+Y_8, 12+Y_8)$	$\mathcal{U}(2,3)$
9	280	9	$(13+Y_9, 15+Y_9, 16+Y_9)$	$\mathcal{U}(3,4)$
10	370	12	$(10+Y_{10}, 11+Y_{10}, 12+Y_{10})$	$\mathcal{U}(1,2)$

Table 2
Fuzzy random demands.

Customer j	t_j	Demand D_j	Parameter Z_j
1	24	$(20+Z_1, 22+Z_1, 23+Z_1)$	$\mathcal{U}(1,2)$
2	22	$(18+Z_2, 20+Z_2, 21+Z_2)$	$\mathcal{U}(1,3)$
3	28	$(16+Z_3, 18+Z_3, 19+Z_3)$	$\mathcal{U}(2,4)$
4	26	$(22+Z_4, 23+Z_4, 24+Z_4)$	$\mathcal{U}(2,3)$
5	19	$(20+Z_5, 22+Z_5, 23+Z_5)$	$\mathcal{U}(3,4)$

and

$$\left.\begin{array}{ll} \mathcal{R}(\mathrm{x},\mathrm{s},\boldsymbol{\xi}(\omega,\gamma)) = \max & \sum_{i=1}^{10}\sum_{j=1}^{5}(r_j - t_{ij} - V_i(\omega,\gamma))y_{ij} \\ \text{subject to} & \sum_{i=1}^{10} y_{ij} \leqslant D_j(\omega,\gamma), \quad j=1,2,\ldots,5, \\ & \sum_{j=1}^{5} y_{ij} \leqslant s_i x_i, \quad i=1,2,\ldots,10, \\ & y_{ij} \geqslant 0, \quad i=1,2,\ldots,10,\ j=1,2,\ldots,5. \end{array}\right\} \tag{26}$$

In problem (24)–(26), the demand-cost vector $\xi = (D_1,\ldots,D_5,V_1,V_2,\ldots,V_{10})$ is a continuous fuzzy random vector, to determine the value of objective function $\mathrm{VaR}_{1-\beta}(\mathrm{x},\mathrm{s})$ with the continuous fuzzy random parameters, the approximation algorithm to fuzzy random VaR (Algorithms 2, 3 in Appendix A) is utilized. In the approximation algorithm, for any feasible solution (x,s), we first generate 5000 random sample points $\hat{\omega}_i, i=1,2,\ldots,5000$, for the random simulation (31) in Appendix A. For each ω_i, we set $l=400$ to generate the fuzzy samples $\hat{\xi}_l^j$, hence from (29) in Appendix A, the error $\mathcal{E}$ can be controlled and it does not exceed $\sqrt{15}/400 \approx 0.01$. Based on the generated samples, for each λ and random sample $\hat{\omega}_i$,

$$\mathrm{Q}_{\hat{\omega}_i}(\mathrm{x},\mathrm{s},\lambda) = \mathrm{Cr}\left\{\sum_{i=1}^{10} c_i x_i - \mathcal{R}(\mathrm{x},\mathrm{s},\boldsymbol{\zeta}_l(\hat{\omega}_i)) \geqslant \lambda\right\}$$

is calculated by (30) in Appendix A, where $\mathcal{R}(\mathrm{x},\mathrm{s},\boldsymbol{\zeta}_l(\hat{\omega}_i))$ (second-stage programming (26)) is determined by the simplex algorithm. Furthermore, for each given λ,

$$\mathcal{Q}(\mathrm{x},\mathrm{s},\lambda) = \int_{\Omega} \mathrm{Q}_{\omega}(\mathrm{x},\mathrm{s},\lambda)\mathrm{Pr}(\mathrm{d}\omega)$$

is computed by random simulation (31) in Appendix A. Finally, we determine the value of $\mathrm{VaR}_{1-\beta}(\mathrm{x},\mathrm{s})$ by an iteration of varying the value of λ in a Bisection Method (Algorithm 2 in Appendix A).

The hybrid MPSO algorithm (Algorithm 1) which fuses the above approximation algorithm is run to solve this VaR-based fuzzy random facility location problem (24)–(26). In the hybrid MPSO, we set the population size $P_{size} = 20$, and run the hybrid algorithm (Algorithm 1) with 200 generations for different confidence levels of 0.9, 0.85, and 0.8. The optimal solutions with different parameters are listed in Table 3, where the *relative error* is given in the last column, which is defined by

$$\mathrm{Error} = \left|\frac{\text{optimal value-objective value}}{\text{optimal value}}\right| \times 100\%.$$

It follows from Table 3 that the relative error does not exceed 1.10%, 1.43% and 1.90% for the different confidence levels $1-\beta = 0.9, 1-\beta = 0.85$, and $1-\beta = 0.8$, respectively, when different parameters are selected. In addition, the convergence

Table 3
Results of hybrid MBACO algorithm with Different Parameters.

No.	System parameters			Results		Error (%)
	$1-\beta$	$P_{m,L}$	$P_{m,C}$	Optimal solution	Objective	
1	0.90	0.2	0.4	$(\langle 0,0\rangle, \langle 1,220.0\rangle, \langle 0,0\rangle, \langle 1,67.4\rangle, \langle 1,185.0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,200.3\rangle, \langle 0,0\rangle)$	−285.8	0.14
2	0.90	0.3	0.3	$(\langle 0,0\rangle, \langle 1,97.7\rangle, \langle 0,0\rangle, \langle 1,119.1\rangle, \langle 1,142.3\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,51.3\rangle, \langle 0,0\rangle)$	−284.0	0.77
3	0.90	0.4	0.2	$(\langle 0,0\rangle, \langle 1,127.0\rangle, \langle 0,0\rangle, \langle 1,204.0\rangle, \langle 1,182.4\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,234.3\rangle, \langle 0,0\rangle)$	−283.0	1.10
4	0.90	0.2	0.3	$(\langle 0,0\rangle, \langle 1,148.8\rangle, \langle 0,0\rangle, \langle 1,95.6\rangle \langle 1,121.2\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,49.5\rangle, \langle 0,0\rangle)$	−286.2	0.00
5	0.90	0.3	0.4	$(\langle 0,0\rangle, \langle 1,218.7\rangle, \langle 0,0\rangle, \langle 1,148.0\rangle, \langle 1,188.2\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,159.7\rangle, \langle 0,0\rangle)$	−284.6	0.56
6	0.85	0.2	0.4	$(\langle 0,0\rangle, \langle 1,92.8\rangle, \langle 0,0\rangle, \langle 1,100.9\rangle, \langle 1,260.0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,247.8\rangle, \langle 0,0\rangle)$	−294.6	0.44
7	0.85	0.3	0.3	$(\langle 0,0\rangle, \langle 1,91.7\rangle, \langle 0,0\rangle, \langle 1,46.7\rangle, \langle 1,180.2\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,147.4\rangle, \langle 0,0\rangle)$	−292.5	1.15
8	0.85	0.4	0.2	$(\langle 0,0\rangle, \langle 1,111.7\rangle, \langle 0,0\rangle, \langle 1,208.8\rangle, \langle 1,239.7\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,177.6\rangle, \langle 0,0\rangle)$	−295.9	0.00
9	0.85	0.2	0.3	$(\langle 0,0\rangle, \langle 1,138.4\rangle, \langle 0,0\rangle, \langle 1,175.3\rangle, \langle 1,54.6\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,91.8\rangle, \langle 0,0\rangle)$	−291.7	1.43
10	0.85	0.3	0.4	$(\langle 0,0\rangle, \langle 1,187.4\rangle, \langle 0,0\rangle, \langle 1,147.5\rangle, \langle 1,138.9\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,186.8\rangle, \langle 0,0\rangle)$	−292.6	1.14
11	0.80	0.2	0.4	$(\langle 0,0\rangle, \langle 1,172.7\rangle, \langle 0,0\rangle, \langle 1,144.0\rangle, \langle 1,199.3\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,105.4\rangle, \langle 0,0\rangle)$	−310.2	0.00
12	0.80	0.3	0.3	$(\langle 0,0\rangle, \langle 1,215.7\rangle, \langle 0,0\rangle, \langle 1,167.1\rangle, \langle 1,173.9\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,54.6\rangle, \langle 0,0\rangle)$	−308.8	0.45
14	0.80	0.4	0.2	$(\langle 0,0\rangle, \langle 1,49.3\rangle, \langle 0,0\rangle, \langle 1,266.0\rangle, \langle 1,250.7\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,241.5\rangle, \langle 0,0\rangle)$	−304.0	1.90
14	0.80	0.2	0.3	$(\langle 0,0\rangle, \langle 1,160.3\rangle, \langle 0,0\rangle, \langle 1,183.1\rangle, \langle 1,260.2\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,260.0\rangle, \langle 0,0\rangle)$	−309.0	0.39
15	0.80	0.3	0.4	$(\langle 0,0\rangle, \langle 1,212.3\rangle, \langle 0,0\rangle, \langle 1,287.3\rangle, \langle 1,145.6\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,160.2\rangle, \langle 0,0\rangle)$	−306.3	1.26

of the best objective value at difference confidence levels is shown in Fig. 3. The performance implies the hybrid MBACO algorithm is robust to the parameter settings when dealing with the VaR-FRFLM.

Recall that in the literature, some hybrid algorithms based on regular particle swarm optimization (PSO)(see [30,31,33]), and genetic algorithm (GA) (see [20,35,36]) have been designed to solve different fuzzy random optimization problems. Those existing approaches cannot be applied to our two-stage VaR-FRFLM, which is a mixed 0–1 integer two-stage fuzzy random programming problem.

Nevertheless, in order to further evaluate the performance of our proposed hybrid MPSO algorithm for the two-stage VaR-FRFLM, we design some other hybrid approaches, that is a hybrid PSO by combining the approximation algorithm to VaR with the regular continuous-binary PSO mechanism and a hybrid GA by integrating the approximation algorithm to VaR and a GA for mixed variables. Then we compare the experimental results of the three different approaches. Here, the population size is set as 20 and the maximum generation is set as 200. In addition, the best mutation and crossover rates (P_m, P_C) are selected for hybrid GA after a number of experiments, where $(P_m, P_C) = (0.3, 0.3)$ for confidence level of 0.9, $(P_m, P_C) = (0.4, 0.2)$ for confidence level of 0.85, and $(P_m, P_C) = (0.4, 0.2)$ for confidence level of 0.80; and in the PSO approach, the learning rates are set as the same values as in (16) and (17). Applying the different approaches to the problem (24)–(26), we obtain the comparison results in Table 4. Furthermore, the convergence comparison is provided in Figs. 4–6, from which we can see that the performance of the hybrid MPSO and PSO are much better (they descend much faster) than the hybrid GA in the early iterations, and the hybrid MPSO outperforms the other two algorithms throughout all the generations. In addition, the location–capacity comparison is provided in Figs. 7–9. From all the above comparison results, we can see the hybrid MPSO algorithm outperforms the other approaches when dealing with the two-stage VaR-FRFLM.

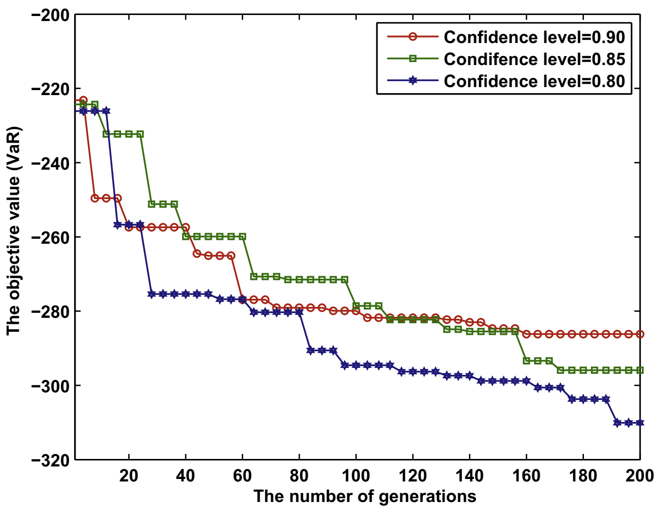

Fig. 3. Convergence of the hybrid MPSO algorithm at different confidence levels.

Table 4
The comparison results of different approaches.

Approach	Optimal solution	Objective value
$1-\beta = 0.9$		
Hybrid MPSO	$(\langle 0,0\rangle, \langle 1,148.8\rangle, \langle 0,0\rangle, \langle 1,95.6\rangle, \langle 1,121.2\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,49.5\rangle, \langle 0,0\rangle)$	−286.2
Hybrid PSO	$(\langle 0,0\rangle, \langle 1,160.3\rangle, \langle 0,0\rangle, \langle 1,100.2\rangle, \langle 1,180.7\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,279.8\rangle, \langle 0,0\rangle)$	−280.1
Hybrid GA	$(\langle 0,0\rangle, \langle 1,173.5\rangle, \langle 0,0\rangle, \langle 1,199.5\rangle, \langle 1,111.3\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,145.2\rangle, \langle 1,102.2\rangle)$	−276.4
$1-\beta = 0.85$		
Hybrid MPSO	$(\langle 0,0\rangle, \langle 1,138.4\rangle, \langle 0,0\rangle, \langle 1,175.3\rangle, \langle 1,54.6\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,91.8\rangle, \langle 0,0\rangle)$	−295.9
Hybrid PSO	$(\langle 0,0\rangle, \langle 1,125.1\rangle, \langle 0,0\rangle, \langle 1,200.0\rangle, \langle 1,180.0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,160.2\rangle, \langle 0,0\rangle)$	−285.1
Hybrid GA	$(\langle 0,0\rangle, \langle 1,163.5\rangle, \langle 0,0\rangle, \langle 1,186.6\rangle, \langle 1,86.5\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,255.8\rangle, \langle 1,74.5\rangle)$	−283.6
$1-\beta = 0.8$		
Hybrid MPSO	$(\langle 0,0\rangle, \langle 1,172.7\rangle, \langle 0,0\rangle, \langle 1,144.0\rangle, \langle 1,199.3\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,105.4\rangle, \langle 0,0\rangle)$	−310.2
Hybrid PSO	$(\langle 0,0\rangle, \langle 1,160.1\rangle, \langle 0,0\rangle, \langle 1,183.3\rangle\langle 1,259.7\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,160.0\rangle, \langle 0,0\rangle)$	−299.4
Hybrid GA	$(\langle 0,0\rangle, \langle 1,129.1\rangle, \langle 0,0\rangle, \langle 1,232.7\rangle, \langle 1,171.7\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \langle 1,127.5\rangle, \langle 0,0\rangle)$	−299.6

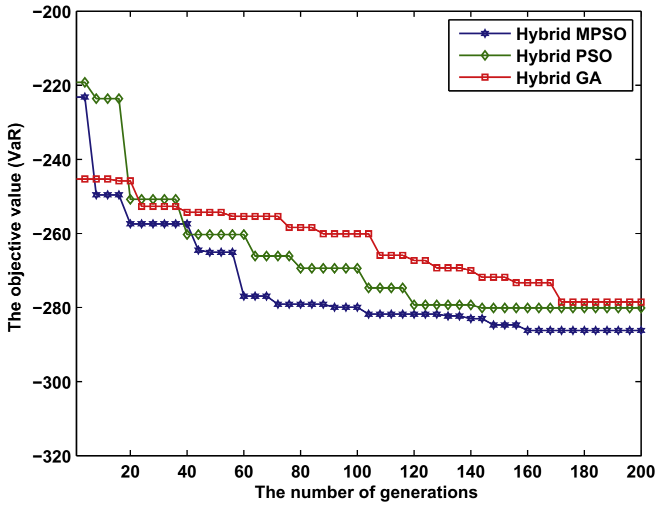

Fig. 4. The convergence comparison of different approaches when $1-\beta = 0.90$.

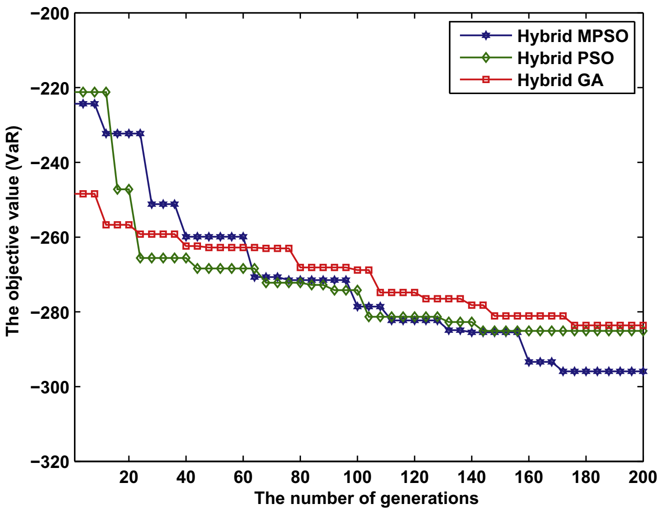

Fig. 5. The convergence comparison of different approaches when $1-\beta = 0.85$.

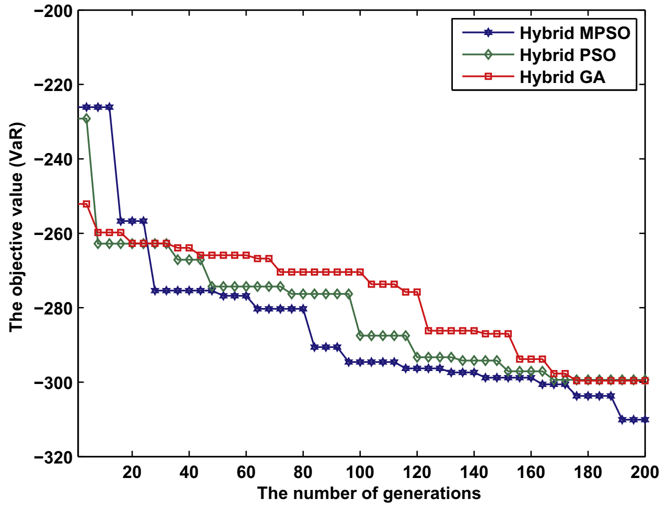

Fig. 6. The convergence comparison of different approaches when $1-\beta=0.80$.

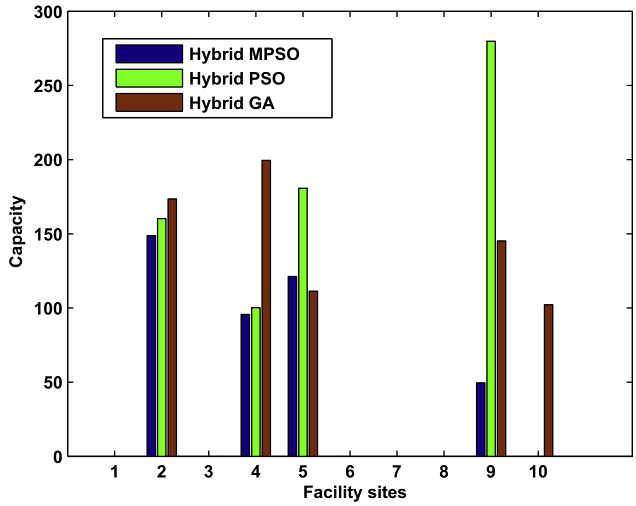

Fig. 7. Location-capacity comparison of different approaches when $1-\beta=0.90$.

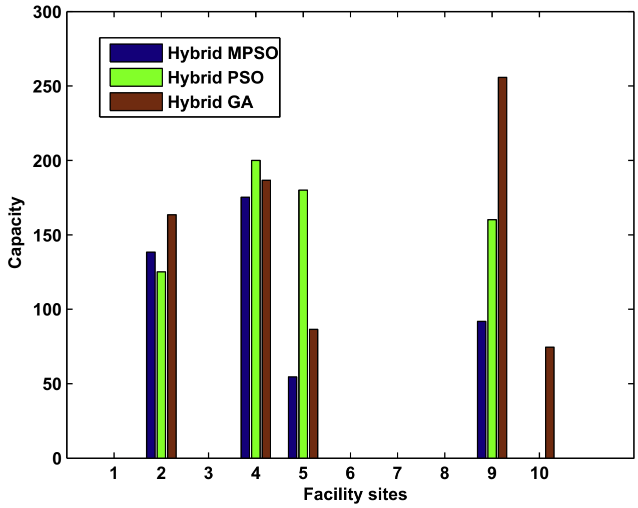

Fig. 8. Location-capacity comparison of different approaches when $1-\beta=0.85$.

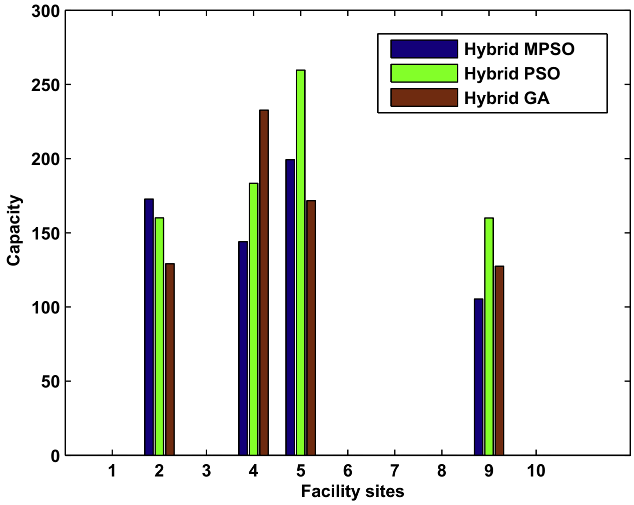

Fig. 9. Location-capacity comparison of different approaches when $1-\beta=0.80$.

6. Conclusions

This paper built a Value-at-Risk-based facility location model with variable capacity and fuzzy random demands and costs. The proposed model is inherently a two-stage mixed 0–1 integer fuzzy random programming problem.

On the one hand, because the model contains continuous fuzzy random parameters, the objective function (fuzzy random VaR) cannot be determined analytically, and the general mixed integer programming methods therefore are not applicable to this new location model. On the other hand, in contrast to the existing fuzzy random location problems in the literature, the capacity in the new model is not fixed but assumed as a decision to be made. This makes our model more practical, but also more complicated than the existing models because, it contain not only continuous decision variables but also binary ones, and they are dependent of each other. Within this structure, the existing approaches to fuzzy random location problems cannot be utilized to handle the VaR-FRFLM either.

To solve the model, a hybrid MPSO algorithm is proposed, in which an approximation algorithm is utilized to compute the fuzzy random VaR, a continuous Nbest–Gbest-based PSO and a genotype-phenotype-based binary PSO vehicles are employed to deal with the continuous capacity decisions and the binary location decisions, respectively, and two mutation operators are implemented to further enhance the search ability of the algorithm. The numerical experiments show that the hybrid MPSO is robust to the parameter settings and exhibits better performance than the hybrid regular continuous-binary PSO and GA approaches.

Based on the current work, the proposed hybrid MPSO algorithm can be generalized to other more mixed 0–1 integer fuzzy random optimization problems with VaR criterion. In addition, the present VaR-based fuzzy random location model can be extended to a multi-objective version by considering the expected profit as an additional objective. The above issues will be discussed in our further studies.

Acknowledgment

This work was supported by the Research Fellowships of the Japan Society for the Promotion of Science (JSPS) for Young Scientists.

Appendix A. Approximation algorithm to fuzzy random VaR

In this Appendix A, we introduce the approximation algorithm to the VaR with continuous fuzzy random parameters, which is proposed in [31]. For the simplicity, here we denote the continuous fuzzy random vector of VaR-FRFLM as $\boldsymbol{\xi}=(\xi_1,\xi_2,\ldots,\xi_{m+n})$ instead of $\boldsymbol{\xi}=(D_1,\ldots,D_m,V_1,\ldots,V_n)$. Therefore, the random parameter involved in $\boldsymbol{\xi}$ is a continuous random vector, and for any $\omega\in\Omega$, fuzzy vector $\xi(\omega)$ is also a continuous one. Let Ξ be the infinite support of ξ as

$$\Xi=\prod_{j=1}^{m+n}[a_j,b_j], \tag{27}$$

where $[a_j,b_j]$ is the support of $\xi_j, j=1,2,\ldots,m+n$. The approximation algorithm to $\mathrm{VaR}_{1-\beta}(\mathrm{x},\mathrm{s})$ with fuzzy random parameter $\boldsymbol{\xi}$ includes the following procedures i–iv.

(i) For the ξ with infinite realizations, we first utilize the discretization method [21] to produce a sequence of fuzzy random vectors $\{\zeta_l\}$ which converges uniformly to ξ, where for any $\omega \in \Omega$, $\zeta_l(\omega)$ is a discrete fuzzy vector.
Given an integer l, we construct the fuzzy random vector $\zeta_l = (\zeta_{l,1}, \zeta_{l,2}, \ldots, \zeta_{l,m+n})$ as follows: for each $1 \leqslant j \leqslant m+n$ we define $\zeta_{l,j} = g_{l,j}(\xi_j)$, for $1 \leqslant j \leqslant m+n$, where $g_{l,j}(\cdot)$'s are given as follows

$$g_{l,j}(u_j) = \sup\left\{\frac{k}{l}\middle| k \in Z, \quad \text{s.t. } \frac{k}{l} \leqslant u_j\right\}, \quad u_j \in [a_j, b_j], \tag{28}$$

Z is the set of integers.By the definition of ζ_l, for any $\omega \in \Omega$, when fuzzy variable $\xi_j(\omega)$ takes on its values in $[a_j, b_j]$, $\zeta_{l,j}(\omega)$ takes on values $\frac{k}{l}$ for $k = [l \cdot a_j], [l \cdot a_j] + 1, \ldots, [l \cdot b_j]$ with $[r]$ the integer part of real number r. In addition, for each k, as $\xi_j(\omega)$ takes on its values in $[\frac{k}{l}, \frac{k+1}{l})$, fuzzy random variable $\zeta_{l,j}(\omega)$ takes on the value $\frac{k}{l}$ only. Consequently, for each $(\omega, \gamma) \in \Omega \times \Gamma$, we have

$$\left|\zeta_{l,j}(\omega)(\gamma) - \xi_j(\omega)(\gamma)\right| < \frac{1}{l}, \quad j = 1, 2, \ldots, m+n.$$

Therefore,

$$\|\zeta_l(\omega)(\gamma) - \xi(\omega)(\gamma)\| = \sqrt{\sum_{j=1}^{m+n} (\zeta_{l,j}(\omega)(\gamma) - \xi_j(\omega)(\gamma))^2} \leqslant \frac{\sqrt{m+n}}{l} \tag{29}$$

for all $(\omega, \gamma) \in \Omega \times \Gamma$, which implies that the sequence $\{\zeta_l\}$ of fuzzy random vectors converges uniformly to the continuous fuzzy random vector ξ. The sequence $\{\zeta_l\}$ is referred to as the discretization of the continuous fuzzy random vector ξ in the rest of the paper.After producing the discretization $\{\zeta_l\}$, given a positive integer l, we denote

$$\mathrm{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}, \zeta_l) = \sup\left\{\lambda |\ \mathrm{Ch}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \zeta_l) \geqslant \lambda\right\} \geqslant \beta\right\}.$$

In the following, we elaborate the approximation of $\mathrm{VaR}_\beta(\mathrm{x})$ through $\mathrm{VaR}_\beta(\mathrm{x}, \zeta_l)$ for each feasible $\boldsymbol{x}$.

(ii) For any $\omega \in \Omega$, the discrete fuzzy vector $\zeta_l(\omega)$ takes on a finite number of values as follows

$$\hat{\zeta}_l^k(\omega) = \left(\hat{\zeta}_{l,1}^k(\omega), \hat{\zeta}_{l,2}^k(\omega), \ldots, \hat{\zeta}_{l,m+n}^k(\omega)\right),$$

with membership degree $\mu_{\omega,k}$, for $k = 1, 2, \ldots, N_\omega$. Therefore, for any $\lambda \in \Re$, we can calculate

$$Q_\omega(\mathrm{x}, \mathrm{s}, \lambda) = \mathrm{Cr}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \zeta_l(\omega)) \geqslant \lambda\right\} = \frac{1}{2}\left[\max_{j=1}^{N_\omega}\{\mu_{\omega,j} | \mathcal{L}_\omega^j(\mathrm{x}, \mathrm{s}) \geqslant \lambda\} + 1 - \max_{j=1}^{N_\omega}\{\mu_{\omega,j} | \mathcal{L}_\omega^j(\mathrm{x}, \mathrm{s}) < \lambda\}\right], \tag{30}$$

where $\mathcal{L}_\omega^j(\mathrm{x}, \mathrm{s}) = \sum_{i=1}^n c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \hat{\zeta}_l^j(\omega))$ is obtained through solving the second-stage programming (6). Here, each second-stage programming is solved by the Simplex Algorithm.

(iii) Denote $Q(\mathrm{x}, \mathrm{s}, \lambda) = \ \mathrm{Ch}\{\sum_{i=1}^n c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \zeta_l) \geqslant \lambda\}$. We generate N random simples $\omega_1, \omega_2, \cdots, \omega_N$ uniformly from the distribution of the random vector ω. Thus

$$\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda) = \mathrm{Ch}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \zeta_l) \geqslant \lambda\right\} = \int_\Omega \mathrm{Cr}\left\{\sum_{i=1}^{n} c_i x_i - \mathcal{R}(\mathrm{x}, \mathrm{s}, \zeta_l(\omega)) \geqslant \lambda\right\}\mathrm{Pr}(\mathrm{d}\omega) = \int_\Omega Q_\omega(\mathrm{x}, \mathrm{s}, \lambda)\mathrm{Pr}(\mathrm{d}\omega)$$

can be computed by the following random simulation

$$\frac{1}{N}\sum_{i=1}^{N} Q_{\omega_i}(\mathrm{x}, \mathrm{s}, \lambda) \xrightarrow{a.s.} \mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda) \quad (N \to \infty). \tag{31}$$

The above random simulation (31) reaches the convergence with probability 1, which is ensured by the strong law of large numbers.

(iv) We note that $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda)$ is a nonincreasing real function of λ, therefore, for each feasible $\boldsymbol{x}$,

$$\mathrm{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}, \zeta_l) = \sup\{\lambda | \mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda) \geqslant \beta\}$$

can be calculated by a Bisection Method.

Summarizing the above procedures **i-iv**, the approximation algorithm can be outlined as follows.

Algorithm 2 [31] Approximation to VaR.

Step 1. Set $\lambda_L = e$ and $\lambda_R = E$, where e and E are a small and a large positive numbers, respectively.
Step 2. Let $\lambda_C = (\lambda_L + \lambda_R)/2$. Calculate $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda_C)$ by calling Algorithm 3.
Step 3. If $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda_C) \geqslant \beta$, then set $\lambda_L = \lambda_C$; otherwise, $\lambda_R = \lambda_C$.

Step 4. Repeat Steps 2–3 until $\lambda_R - \lambda_L < \varepsilon$, where $\varepsilon > 0$ is a prescribed sufficiently small number.
Step 5. Return the value of $\mathrm{VaR}_{1-\beta}(\mathrm{x}, \mathrm{s}, \zeta_l) = (\lambda_L + \lambda_R)/2$.

In Algorithm 2, the computation of $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda)$ is given by the following algorithm, which bases the procedures i–iii.

Algorithm 3 [31] Computing $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda)$.

Step 1. Set $Q = 0$ and $l = L$, where L is a sufficient large integer.
Step 2. Randomly generate a simple point $\hat{\omega}$ from the distribution of the continuous random vector ω.
Step 3. Generate simples $\hat{\zeta}_l^k(\hat{\omega}) = (\hat{\zeta}_{l,1}^k(\hat{\omega}), \hat{\zeta}_{l,2}^k(\hat{\omega}), \ldots, \hat{\zeta}_{l,m+n}^k(\hat{\omega}))$ through (28) from the support Ξ of ξ for $k = 1, 2, \ldots, N_{\hat{\omega}}$.
Step 4. Calculate $\mathcal{L}_{\hat{\omega}}^k(\mathrm{x}, \mathrm{s})$ by solving the second-stage programming (6) with the simplex algorithm for $k = 1, 2, \ldots, N_{\hat{\omega}}$.
Step 5. Calculate $Q_{\hat{\omega}}(\mathrm{x}, \mathrm{s}, \lambda)$ via (30).
Step 6. $Q \leftarrow Q + Q_{\hat{\omega}}(\mathrm{x}, \mathrm{s}, \lambda)$.
Step 7. Repeat the Steps 2–6 N times.
Step 8. Return the value of $\mathcal{Q}(\mathrm{x}, \mathrm{s}, \lambda) = Q/N$.

References

[1] O. Berman, Z. Drezner, A probablistic one-centre location problem on a network, Journal of the Operational Research Society 54 (8) (2003) 871–877.
[2] U. Bhattacharya, J.R. Rao, R.N. Tiwari, Fuzzy multi-criteria facility location problem, Fuzzy Sets and Systems 51 (3) (1992) 277–287.
[3] E. Carrizosa, E. Conde, M. Munoz-Marquez, J. Puerto, The generalized Weber problem with expected distances, Rairo–Recherche Operationnelle-Operations Research 29 (1) (1995) 35–57.
[4] M. Clerc, J. Kennedy, The particle swarm – Explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.
[5] D. Dubois, H. Prade, Possibility Theory, Plenum Press, New York, 1988.
[6] D. Duffie, J. Pan, An overview of value-at-risk, Journal of Derivatives 4 (3) (1997) 7–49.
[7] H. Ishii, Y.L. Lee, K.Y. Yeh, Fuzzy facility location problem with preference of candidate sites, Fuzzy Sets and Systems 158 (17) (2007) 1922–1930.
[8] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the 1995, IEEE International Conference on Neural Networks IV, 1995, pp. 1942–1948.
[9] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, 1997, pp. 4104–4108.
[10] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.
[11] R. Kruse, K.D. Meyer, Statistics with Vague Data, D. Reidel Publishing Company, Dordrecht, 1987.
[12] H. Kwakernaak, Fuzzy random variables – I. Definitions and theorems, Information Sciences 15 (1) (1978) 1–29.
[13] M. Shen, Y. Tsai, CPW-fed monopole antenna characterized by using particle swarm optimization incorporating decomposed objective functions, International Journal of Innovative Computing, Information and Control 4 (8) (2008) 1897–1919.
[14] G. Laporte, F.V. Louveaux, L.V. Hamme, Exact solution to a location problem with stochastic demands, Transportation Science 28 (2) (1994) 95–103.
[15] S. Lee, S. Soak, S. Oh, W. Pedrycz, M. Jeon, Modified binary particle swarm optimization, Progress in Natural Science 18 (9) (2008) 1161–1166.
[16] L. Lin, H.M. Lee, A fuzzy decision support system for facility site selection of multinational enterprises, International Journal of Innovative Computing, Information and Control 3 (1) (2007) 151–162.
[17] B. Liu, Uncertainty Theory, second ed., Springer-Verlag, Berlin, 2007.
[18] B. Liu, Y.K. Liu, Expected value of fuzzy variable and fuzzy expected value models, IEEE Transaction on Fuzzy Systems 10 (4) (2002) 445–450.
[19] Y.K. Liu, B. Liu, Fuzzy random variable: a scalar expected value operator, Fuzzy Optimization and Decision Making 2 (2) (2003) 143–160.
[20] Y.K. Liu, B. Liu, On minimum-risk problems in fuzzy random decision systems, Computers and Operations Research 32 (2) (2005) 257–283.
[21] Y.K. Liu, The approximation method for two-stage fuzzy random programming with recourse, IEEE Transactions on Fuzzy Systems 15 (6) (2007) 1197–1208.
[22] R. Logendran, M.P. Terrell, Uncapacitated plant location allocation problems with price sensitive stochastic demands, Computers and Operations Research 15 (2) (1988) 189–198.
[23] F.V. Louveaux, D. Peeters, A dual-based procedure for stochastic facility location, Operations Research 40 (3) (1992) 564–573.
[24] M. López-Diaz, M.A. Gil, Constructive definitions of fuzzy random variables, Statistics and Probability Letters 36 (2) (1997) 135–143.
[25] W. Pedrycz, F. Gomide, Fuzzy Systems Engineering: Toward Human-Centric Computing, J. Wiley, Hoboken, NJ, 2007.
[26] W. Pedrycz, Statistically grounded logic operators in fuzzy sets, European Journal of Operational Research 193 (2) (2009) 520–529.
[27] W. Pedrycz, B.J. Park, N.J. Pizzi, Identifying core sets of discriminatory features using particle swarm optimization, Expert Systems with Applications 36 (3) (2009) 4610–4616.
[28] A. Ratnweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 240–255.
[29] R. Schutz, L. Stougie, A. Tomasgard, Stochastic facility location with general long-run costs and convex short-run cost, Computers and Operations Research 35 (9) (2008) 2988–3000.
[30] S. Wang, J. Watada, Reliability optimization of a series–parallel system with fuzzy random lifetimes, International Journal of Innovative Computing, Information and Control 5 (6) (2009) 1547–1558.
[31] S. Wang, J. Watada, Value-at-Risk-based fuzzy stochastic optimization problems, in: Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju Island, Korea, 2009.
[32] S. Wang, J. Watada, W. Pedrycz, Value-at-Risk-based two-stage fuzzy facility location problems, IEEE Transactions on Industrial Informatics 5 (4) (2009) 465–482.
[33] S. Wang, J. Watada, W. Pedrycz, Fuzzy random facility location problems with recourse, in: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, Texas, USA, 2009.
[34] M. Wen, K. Iwamura, Fuzzy facility location-allocation problem under the Hurwicz criterion, European Journal of Operational Research 184 (2) (2008) 627–635.
[35] M. Wen, K. Iwamura, Facility location-allocation problem in random fuzzy environment: Using (α, β)-cost minimization model under the Hurewicz criterion, Computers and Mathematics with Applications 55 (4) (2008) 704–713.
[36] L. Yang, Chance-constrained methods for optimization problems with random and fuzzy parameters, International Journal of Innovative Computing, Information and Control 5 (2) (2009) 413–422.

[37] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1) (1978) 3–28.
[38] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) – an outline, Information Sciences 172 (1-2) (2005) 1–40.
[39] L.A. Zadeh, Is there a need for fuzzy logic?, Information Sciences 178 (13) (2008) 2751–2779
[40] T. Zeng, J. Ward, The stochastic location-assignment problem on a tree, Annals of Operations Research 136 (1) (2005) 81–97.
[41] J. Zhou, B. Liu, Modeling capacitated location-allocation problem with fuzzy demands, Computers and Industrial Engineering 53 (3) (2007) 454–468.