J Intell Manuf (2015) 26:447–458
DOI 10.1007/s10845-013-0801-7

Designing fuzzy supply chain network problem by mean-risk optimization method

Guoqing Yang · Yankui Liu

Received: 22 August 2012 / Accepted: 3 June 2013 / Published online: 15 June 2013

Abstract In many practical supply chain network design (SCND) problems, the critical parameters such as customer demands, transportation costs and resource capacities are quite uncertain. The significance of uncertainty motivates us to develop a new mean-risk fuzzy optimization method for SCND problem, in which the standard semivariance is suggested to gauge the risk resulted from fuzzy uncertainty. To demonstrate the advantages of the proposed optimization method, we define a new concept about the value of fuzzy solution for our SCND problem. When the transportation costs and the demands of customers have continuous possibility distributions, we approximate the continuous fuzzy vector by a sequence of discrete fuzzy vectors. On the basis of the approximation scheme, we obtain an approximating optimization model, which is a nonlinear mixed-integer programming problem. Furthermore, we design a hybrid memetic algorithm (MA) to solve the approximating optimization problem. The designed hybrid MA incorporates the reduced variable neighborhood search to act as the local search procedure. Finally, we conduct some numerical experiments via an application example to demonstrate the effectiveness of the designed hybrid MA.

Keywords Supply chain network design · Fuzzy optimization · Risk measure · Approximation method · Hybrid memetic algorithm

G. Yang · Y. Liu (✉)
Risk Management & Financial Engineering Laboratory, College of Mathematics & Computer Science, Hebei University, Baoding 071002, Hebei, China
e-mail: yliu@hbu.edu.cn

G. Yang
e-mail: ygqfq100@gmail.com

Introduction

A supply chain is a network of suppliers, manufacturing plants, warehouses, and distribution channels organized to acquire raw materials, convert these raw materials to finished products, and distribute these products to customers. The SCND problem is to make the decisions to satisfy the demands of customers and minimize the sum of strategic and tactical costs. The goal of the SCND problem is to provide an optimal platform for efficient and effective supply chain management. Geoffrion and Graves (1974) first described a comprehensive mixed-integer programming model for the design of supply chain networks. This model was later extended to incorporate more realistic issues in production, warehousing and distribution. For the development of deterministic SCND problems, the interested reader may refer to Aikens (1985), Cohen and Lee (1989), Geoffrion and Powers (1995), Alonso-Ayuso et al. (2003), Bachlaus et al. (2008), Bidhandi et al. (2009), Cho and Lee (2012).

In many practical SCND problems, the critical parameters such as customer demands, transportation costs and resource capacities are quite uncertain. The significance of uncertainty has motivated some researchers to address stochastic parameters in supply chain problems (Cheung and Powell 1996; Sabri and Beamon 2000; Santoso et al. 2005; Georgiadis et al. 2011; Yang et al. 2011; Mizgiera et al. 2012). In the meantime, a number of researchers have proposed fuzzy optimization approaches to modeling the supply chain problems. For example, Petrovic et al. (1999) used fuzzy logic to interpret imprecise information in an uncertain environment; Giannoccaro et al. (2003) proposed a fuzzy echelon methodology for supply chain inventory management, in which triangular fuzzy numbers were used to model the uncertainty associated with market demand; Wang and Shu (2005) developed a fuzzy supply chain model by using six-point fuzzy numbers; Zhou et al. (2008) considered a fuzzy two-echelon supply chain composed by a manufacturer and a retailer, in which the customer demands and the manufacturing costs were modeled as fuzzy variables; Based on fuzzy analytic network process and preemptive fuzzy integer goal programming, Wong (2012) proposed a decision support system to provide selections of logistics outsourcing providers in a global supply chain; Kubat and Yuce (2012) proposed a general framework, which combines analytic hierarchy process (AHP), fuzzy AHP and genetic algorithm, to determine the best set of suppliers, and Lee et al. (2012) proposed a fuzzy analytic network process model to evaluate various aspects of suppliers.

In this paper, we develop a new two-stage mean-risk fuzzy optimization method for SCND problem. The objective of the proposed optimization model is to minimize the total expected costs and the risk of excessive costs. When the underlying distribution of costs is asymmetric, the semivariance is more useful than the variance, so we employ the standard semivariance of fuzzy variable as a new risk measure in our SCND problem. To show the advantages of the proposed optimization methods, we define a new concept about the value of fuzzy solution for our SCND problem. When the fuzzy parameters have continuous possibility distributions, we cannot obtain the analytical expression of the second-stage value function. Therefore, the proposed two-stage SCND problem cannot be solved by conventional optimization methods. In this case, we approximate the continuous fuzzy vector by a sequence of discrete fuzzy vectors. On the basis of the approximation scheme, we obtain an approximating optimization model, which is a nonlinear mixed-integer programming problem. Furthermore, we design a hybrid MA incorporating the RVNS to solve the approximating SCND problem. The MA is an evolutionary algorithm in which the local search plays a significant role. The MA combines the global search and the local search by using the genetic algorithm to perform the exploration and exploitation. The MA has shown a great record of efficient implementations, and provided good solution results for complex optimization problems (Yeh 2006; Pishvaee et al. 2010; Jolai et al. 2011; Urselmann et al. 2011).

The remainder of this paper is organized as follows. Section "Formulation of mean-risk SCND problem" presents a new two-stage mean-risk SCND problem. Section "The value of fuzzy solution in mean-risk SCND problem" defines a new concept about the value of fuzzy solution for the proposed two-stage mean-risk SCND problem. Section "The approximation method for mean-risk SCND problem" discusses the approximation scheme for the mean-risk SCND problem. Section "Hybrid solution methods" designs a hybrid MA by incorporating the RVNS to solve the approximating optimization problem. Section "Numerical experiments and comparison study" conducts some numerical experiments via an application example to demonstrate the effectiveness of the designed hybrid MA. The conclusions are summarized in the last section.

Formulation of mean-risk SCND problem

In this section, we develop a fuzzy programming approach to modeling the SCND problem. The notations of our SCND problem are listed in Table 1.

In our SCND problem, building plants and warehouses requires some time and decision makers don't know in advance the realizations of fuzzy variables, therefore the decisions about plants and warehouses are made under uncertainty. That is, the decisions (u_j, w_l) are the first-stage decision variables, they are taken before knowing the values of uncertain costs and demands. After knowing the values of fuzzy parameters, the decisions x_{ij}, y_{jlk} and z_{lmk} are made in the second stage, and they are referred to as the second-stage decision variables.

Constraints (1) ensure that the raw material to plant j meets the need of the plant, and the amount of products from warehouse l to customers is equal to the amount of products from plants to the warehouse l,

$$\begin{array}{ll} \sum_{i\in I} x_{ij} = \sum_{k\in K} r_k \sum_{l\in L} y_{jlk}, & \forall j \in J, \\ \sum_{j\in J} y_{jlk} = \sum_{m\in M} z_{lmk}, & \forall l \in L, \forall k \in K. \end{array} \tag{1}$$

For each realization γ of fuzzy demand d_{mk}, the total amount of product k from warehouses to customer m meets the demand $d_{mk}(\gamma)$ of the customer. If there is some shortfall for the demand of the customer, then we introduce an additional term to penalize this shortfall. This situation is expressed as constraints (2),

$$\sum_{l\in L} z_{lmk} + \tilde{z}_{mk} \geq d_{mk}(\gamma), \quad \forall m \in M, \forall k \in K. \tag{2}$$

During the production process, the plants and warehouses have their capacity, and the products are less than the quantity. We model the capacity limitations by constraints (3),

$$\begin{array}{ll} \sum_{j\in J} x_{ij} \leq s_i, & \forall i \in I, \\ \sum_{k\in K} r^P_{jk} \sum_{l\in L} y_{jlk} \leq a_j u_j, & \forall j \in J, \\ \sum_{k\in K} r^L_{lk} \sum_{j\in J} y_{jlk} \leq h_l w_l, & \forall l \in L. \end{array} \tag{3}$$

For the sake of presentation, we use $\mathbf{u}$ to denote the first-stage decision vector (u_j, w_l), and $\mathbf{y}$ to denote the second-stage decision vector $(x_{ij}, y_{jlk}, z_{lmk})$. Fuzzy vector $(cp_{ij}(\gamma), cp'_{jlk}(\gamma), cp''_{lmk}(\gamma), d_{mk}(\gamma))$ is represented by ξ. We sometimes denote $\mathbf{y}$ as $\mathbf{y}(\gamma)$ to represent that $\mathbf{y}$ depends on γ. However, the dependence of $\mathbf{y}$ on γ is completely different from the dependence of ξ on γ. The second-stage

Table 1 List of notations

Notations	Definitions
Sets and indexes	
I	The index set of suppliers
J	The index set of plants
L	The index set of warehouses
K	The index set of products
M	The index set of customers
Constant parameters	
cm_j	The cost of building the plant j
cw_l	The cost of building the warehouse l
cq_i^I	The unit cost of raw material from supplier i
cq_{jk}^J	The unit production cost of product k in plant j
p_{mk}	The unit penalty cost of product k for customer m
s_i	The capacity of raw material for supplier i
h_l	The storage capacity in warehouse l
a_j	The product capacity in plant j
r_{jk}^P	The processing requirement for per-unit product k in plant j
r_{lk}^L	The processing requirement for per-unit product k in warehouse l
r_k	The raw material required for per-unit product k
Fuzzy parameters	
$cp_{ij}(\gamma)$	The unit transportation cost from supplier i to plant j for raw material
$cp'_{jlk}(\gamma)$	The unit transportation cost from plant j to warehouse l for product k
$cp''_{lmk}(\gamma)$	The unit transportation cost from warehouse l to customer m for product k
$d_{mk}(\gamma)$	The demand of customer m for product k
Decision variables	
x_{ij}	The amount of raw material transported from supplier i to plant j
y_{jlk}	The amount of product k transported from plant j to warehouse l
z_{lmk}	The amount of product k transported from warehouse l to customer m
$\tilde{z}_{mk}$	The shortfall of product k for customer m
u_j	Decision to build or not to build the plant j
w_l	Decision to build or not to build the warehouse l

decision vector $\mathbf{y}(\gamma)$ is not functional but simply indicates that the decisions $\mathbf{y}$ are typically not the same under different realizations of γ.

If the first-stage decision vector $\mathbf{u}$ is fixed, and a realization $\xi(\gamma)$ of fuzzy vector ξ is known, then the second-stage programming model for the SCND problem reads

$$
\begin{aligned}
Q(\mathbf{u}, \xi(\gamma)) = \min\ & \sum_{i\in I}\sum_{j\in J} cp_{ij}(\gamma)x_{ij} + \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} cp'_{jlk}(\gamma)y_{jlk} \\
& + \sum_{k\in K}\sum_{l\in L}\sum_{m\in M} cp''_{lmk}(\gamma)z_{lmk} \\
& + \sum_{i\in I} cq_i^I \sum_{j\in J} x_{ij} + \sum_{k\in K}\sum_{j\in J} cq_{jk}^J \sum_{l\in L} y_{jlk} \\
& + \sum_{l\in L}\sum_{m\in M} p_{mk}\tilde{z}_{mk} \\
\text{subject to:}\ & \sum_{i\in I} x_{ij} = \sum_{k\in K} r_k \sum_{l\in L} y_{jlk}, \quad \forall j \in J \\
& \sum_{j\in J} y_{jlk} = \sum_{m\in M} z_{lmk}, \quad \forall l \in L, \forall k \in K \\
& \sum_{l\in L} z_{lmk} + \tilde{z}_{mk} \ge d_{mk}(\gamma), \quad \forall m \in M, \forall k \in K \\
& \sum_{j\in J} x_{ij} \le s_i, \quad \forall i \in I \\
& \sum_{k\in K} r_{jk}^P \sum_{l\in L} y_{jlk} \le a_j u_j, \quad \forall j \in J \\
& \sum_{k\in K} r_{lk}^L \sum_{j\in J} y_{jlk} \le h_l w_l, \quad \forall l \in L \\
& x_{ij} \ge 0,\ y_{jlk} \ge 0,\ z_{lmk} \ge 0, \quad \forall i, j, k, l, m.
\end{aligned} \tag{4}
$$

In problem (4), $Q(\mathbf{u}, \xi(\gamma))$ is the optimal value of second-stage problem (4), and called the recourse cost function. Since $Q(\mathbf{u}, \xi(\gamma))$ is a fuzzy variable, the total cost function

$$f(\mathbf{u},\xi) = \sum_{j\in J} cm_j u_j + \sum_{l\in L} cw_l w_l + Q(\mathbf{u},\xi(\gamma)) \quad (5)$$

is also a fuzzy variable. Therefore, determining the optimal decision vector $\mathbf{u}$ leads to the problem of comparing fuzzy cost variables $\{f(\mathbf{u},\xi)\}_{\mathbf{u}\in\mathbf{U}}$, where U is the set of all feasible first-stage decision vectors. Since we focus on the total costs, smaller values of $f(\mathbf{u},\xi)$ are preferred. While comparing fuzzy variables, it is crucial to consider the effect of variability and specify the preference relations among fuzzy variables using risk measures. We will define a new risk measure ρ, and build the following mean-risk model for our SCND problem,

$$\min_{\mathbf{u}\in\mathbf{U}}\{\boldsymbol{E}_\xi[\mathbf{f}(\mathbf{u},\xi)] + \lambda\rho(\mathbf{f}(\mathbf{u},\xi))\}, \quad (6)$$

where $\boldsymbol{E}_\xi$ is the expected value of fuzzy variable (Liu and Liu 2002).

In problem (6), λ is a non-negative trade-off coefficient representing the exchange rate of mean cost for risk. We also refer to it as the risk coefficient, which is specified by decision makers according to their risk preferences. Particularly, when λ is 0, problem (6) is the risk-neural expected value model (Liu 2005; Sun et al. 2011).

In the following, we specify $\rho(f(\mathbf{u},\xi))$ as the standard semivariance of fuzzy cost $f(\mathbf{u},\xi)$. In this case, the objective function of problem (6) becomes

$$\min_{u\in U}\{\boldsymbol{E}_\xi[f(\mathbf{u},\xi)] + \lambda\sqrt{SV^+[f(\mathbf{u},\xi)]}\}, \quad (7)$$

where the standard semivariance $\sqrt{SV^+[f(\mathbf{u},\xi)]}$ is defined as,

$$\begin{aligned}
&\sqrt{SV^+[f(\mathbf{u},\xi)]} \\
&= \sqrt{\boldsymbol{E}_\xi[((f(\mathbf{u},\xi) - \boldsymbol{E}_\xi[f(\mathbf{u},\xi)])^+)^2]} \\
&= \sqrt{\boldsymbol{E}_\xi\left[\left(\left(\sum_{j\in J} cm_j u_j + \sum_{l\in L} cw_l w_l + Q(\mathbf{u},\xi(\gamma)) - \boldsymbol{E}_\xi[f(\mathbf{u},\xi)]\right)^+\right)^2\right]} \\
&= \sqrt{\boldsymbol{E}_\xi[((Q(\mathbf{u},\xi(\gamma)) - \boldsymbol{E}_\xi[Q(\mathbf{u},\xi(\gamma))])^+)^2]} \\
&= \sqrt{SV^+[Q(\mathbf{u},\xi(\gamma))]}. \qquad (8)
\end{aligned}$$

When X is a fuzzy variable, by the definition of standard semivariance, we have

$$0 \le \sqrt{SV^+[X]} \le \sigma[X],$$

where $\sigma[X] = \sqrt{V[X]}$ is a standard variance, and if X has a symmetric possibility distribution, then we have $\sqrt{SV^+[X]} = \sigma[X]$.

Note that the recourse cost $Q(\mathbf{u},\xi(\gamma))$ is usually an asymmetric fuzzy variable. Therefore, the standard semivariance as a risk criterion is better than variance in our SCND problem.

Finally, we formally build a two-stage mean-risk programming model for SCND problem as follows:

$$\begin{aligned}
\min\ & \sum_{j\in J} cm_j u_j + \sum_{l\in L} cw_l w_l + \boldsymbol{E}_\xi[Q(\mathbf{u},\xi)] \\
& + \lambda\sqrt{SV^+[Q(\mathbf{u},\xi)]} \qquad (9)\\
\text{subject to: } & u_j, w_l \in \{0,1\}, \quad \forall j\in J, \forall l\in L,
\end{aligned}$$

where $Q(\mathbf{u},\xi(\gamma))$ is the optimal value of the following program problem

$$\begin{aligned}
Q(\mathbf{u},\xi(\gamma)) = \min\ & \sum_{i\in I}\sum_{j\in J} cp_{ij}(\gamma)x_{ij} + \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} cp'_{jlk}(\gamma)y_{jlk} \\
& + \sum_{k\in K}\sum_{l\in L}\sum_{m\in M} cp''_{lmk}(\gamma)z_{lmk} \\
& + \sum_{i\in I} cq_i^I \sum_{j\in J} x_{ij} + \sum_{k\in K}\sum_{j\in J} cq_{jk}^J \sum_{l\in L} y_{jlk} \\
& + \sum_{l\in L}\sum_{m\in M} p_{mk}\tilde{z}_{mk} \\
\text{subject to: } & \sum_{i\in I} x_{ij} = \sum_{k\in K} r_k \sum_{l\in L} y_{jlk}, \quad \forall j\in J \\
& \sum_{j\in J} y_{jlk} = \sum_{m\in M} z_{lmk}, \quad \forall l\in L,\ \forall k\in K \\
& \sum_{l\in L} z_{lmk} + \tilde{z}_{mk} \ge d_{mk}(\gamma), \quad \forall m\in M,\ \forall k\in K \\
& \sum_{j\in J} x_{ij} \le s_i, \quad \forall i\in I \\
& \sum_{k\in K} r_{jk}^P \sum_{l\in L} y_{jlk} \le a_j u_j, \quad \forall j\in J \\
& \sum_{k\in K} r_{lk}^L \sum_{j\in J} y_{jlk} \le h_l w_l, \quad \forall l\in L \\
& x_{ij}\ge 0,\ y_{jlk}\ge 0,\ z_{lmk}\ge 0, \quad \forall i,j,k,l,m.
\end{aligned}$$

The objective of problem (9) is to minimize the total costs with minimum semivariance, where the costs involve the total investment and operational costs. The demand-shortage penalty $\sum_{l\in L}\sum_{m\in M} p_{mk}\tilde{z}_{mk}$ guarantees that $Q(\mathbf{u},\xi(\gamma)) < +\infty$ for all $\mathbf{u}$ and $\xi(\gamma)$. Therefore, for a fixed $\mathbf{u}$ and a known fuzzy event γ, problem (4) always has feasible solutions. To solve problem (9), the computation about the expected value and standard semivariance of $Q(\mathbf{u},\xi(\gamma))$ is a challenge issue. In the next section, we define a new concept about the value of fuzzy solution for problem (9), which helps the reader to understand the advantages of the proposed optimization method.

The value of fuzzy solution in mean-risk SCND problem

Since the mean-risk SCND problem (9) is computationally difficult to be solved, some decision makers are inclined to solve its simpler versions. One popular simple version is the deterministic programming problem obtained by replacing all fuzzy variables included in problem (9) by their expected values. A natural question is whether this method is sometimes nearly optimal or whether it is totally inaccurate. The purpose of this section is to give the theoretical answer to this question.

If we replace all fuzzy variables included in problem (9) by their expected values, then we obtain the following expected value problem

$$EV = \min_{\mathbf{u}\in U} f\left(\mathbf{u}, \bar{\xi}\right), \tag{10}$$

where $\bar{\xi} = \boldsymbol{E}(\xi)$ denotes the expected value of fuzzy vector ξ. Let $\bar{\mathbf{u}}(\bar{\xi})$ be an optimal solution of the expected value problem (10), and called the *expected value solution*.

Firstly, the expected result of using the expected value solution in the sense of mean-risk is defined as

$$MREV = \boldsymbol{E}_{\xi}[f\left(\bar{\mathbf{u}}(\bar{\xi}), \xi\right)] + \lambda\sqrt{SV^{+}[f\left(\bar{\mathbf{u}}(\bar{\xi}), \xi\right)]}, \tag{11}$$

where MR is the abbreviation of mean-risk. Therefore, the quantity MREV measures how the decision $\bar{\mathbf{u}}(\bar{\xi})$ performs, allowing the second-stage decision to be chosen optimally as functions of $\bar{\mathbf{u}}(\bar{\xi})$ and ξ.

Secondly, we compare the expected value solution to the so-called *here-and-now solution* corresponding to the recourse problem (RP) defined in (9), and write that as

$$MRRP = \min_{\mathbf{u}\in U}\left\{\boldsymbol{E}_{\xi}[f\left(\mathbf{u}, \xi\right)] + \lambda\sqrt{SV^{+}[f\left(\mathbf{u}, \xi\right)]}\right\}. \tag{12}$$

Finally, we define the *value of fuzzy solution* in the sense of mean-risk as the difference between the here-and-now solution and the expected value solution, i.e.,

$$MRVFS = MREV - MRRP. \tag{13}$$

The following theorem establishes the relation between the MREV and MRRP.

Theorem 1 *For any two-stage mean-risk SCND problem* (9)*, the following relation holds true*

$$MRRP \leq MREV, \tag{14}$$

which implies $MRVFS \geq 0$.

Proof The solution $\bar{\mathbf{u}}(\bar{\xi})$ is just a feasible solution of mean-risk SCND problem (9). Therefore, the mean-risk function value associated with $\bar{\mathbf{u}}(\bar{\xi})$, denoted by MREV, is larger than or equal to MRRP. It follows that $MRVFS \geq 0$. The proof of the theorem is complete.

Theorem 1 indicates that the MRVFS is nonnegative, which gives us the motivation for the solution of two-stage mean-risk SCND problem (9). The MRVFS measures the value of knowing and using the possibility distributions on future realizations. In this paper, our emphasis is on SCND problem where no further information about fuzzy costs and demands is available, so the quantity MRVFS becomes practically relevant to our mean-risk SCND problem.

The approximation method for mean-risk SCND problem

In order to solve problem (9), it is required to compute the expected value $\boldsymbol{E}_{\xi}[Q\left(\mathbf{u}, \xi(\gamma)\right)]$ as well as the standard semi-variance $\sqrt{SV^{+}[Q\left(\mathbf{u}, \xi\right)]}$. For any fixed $\mathbf{u}$ and realized value $\xi(\gamma)$, we solve linear programming problem (4) by simplex algorithm. When ξ is a continuous fuzzy vector, the computation of $\boldsymbol{E}_{\xi}[Q\left(\mathbf{u}, \xi(\gamma)\right)]$ requires to solve an infinite number of linear programming problems in the second stage, which is impossible from the viewpoint of the computation. Therefore, we are required to approximate the continuous fuzzy vector by a sequence of discrete fuzzy vectors. In the following, we address this issue in details.

In our SCND problem, the parameter $cp_{ij}(\gamma)$ is the unit transportation cost of raw material from supplier i to plant j, and it depends on a fuzzy parameter ξ_1 representing the fuel price. Hence, $cp_{ij}(\gamma)$ is a linear function of ξ_1, $cp_{ij}(\gamma) = cp\_a_{ij} \times \xi_1(\gamma) + cp\_b_{ij}$, where cp_a_{ij} and cp_b_{ij} are real numbers and cp_a_{ij} is the distance between supplier i and plant j. Using this method, we denote $cp'_{jlk}(\gamma)$, $cp''_{lmk}(\gamma)$ and $d_{mk}(\gamma)$ by the linear functions of $\xi_2(\gamma)$, $\xi_3(\gamma)$ and $\xi_4(\gamma)$, respectively. Therefore, the parameters $cp_{ij}(\gamma)$, $cp'_{jlk}(\gamma)$, $cp''_{lmk}(\gamma)$ and $d_{mk}(\gamma)$ are the functions of fuzzy vector $\xi = (\xi_1, \xi_2, \xi_3, \xi_4)$. In addition, we assume that the demands of customers and transportation costs are related with each other. To characterize the relations among fuzzy variables ξ_1, ξ_2, ξ_3 and ξ_4, we assume $(\xi_1, \xi_2, \xi_3, \xi_4)$ has the following joint possibility distribution,

$$\pi(x) = \exp\left\{-\frac{1}{2}(x-\mu)^{T}\Sigma(x-\mu)\right\}, \tag{15}$$

where $x = (x_1, x_2, x_3, x_4) \in \prod_{i=1}^{4}[a_i, b_i]$, Σ is a (4×4) positive definite matrix, and $\mu \in R^4$ is a constant vector. In this case, we approximate the continuous fuzzy vector ξ by a sequence $\{\zeta_n\}$ of discrete fuzzy vectors (Liu 2006).

For any given integer n, the discrete fuzzy vector $\zeta_n = (\zeta_{n,1}, \zeta_{n,2}, \zeta_{n,3}, \zeta_{n,4})$ is defined as

$$\zeta_n = g_n\left(\xi\right) = \left(g_{n,1}(\xi_1), g_{n,2}(\xi_2), g_{n,3}(\xi_3), g_{n,4}(\xi_4)\right),$$

where the fuzzy variables $\zeta_{n,i} = g_{n,i}\left(\xi_i\right)$, $n = 1, 2, \ldots, i = 1, 2, 3, 4$, such that

$$g_{n,i}(u_i) = \sup\left\{\frac{k_i}{n} \mid k_i \in Z \text{ such that } \frac{k_i}{n} \leq u_i\right\}, \quad u_i \in [a_i, b_i]$$

with Z being the set of integers. Using the approximation method, fuzzy vector ζ_n takes its values in the following set

$$\left\{\hat{\zeta}_n^q = (\hat{\zeta}_{n,1}^q, \hat{\zeta}_{n,2}^q, \hat{\zeta}_{n,3}^q, \hat{\zeta}_{n,4}^q), q = 1, 2, \ldots, N\right\},$$

and the possibility that ζ_n takes the value $\hat{\zeta}_n^q$ is

$$\mathrm{Pos}\left\{\zeta_n = \hat{\zeta}_n^q\right\} = \mathrm{Pos}\left\{\zeta_{n,i} = \hat{\zeta}_{n,i}^q, i = 1, 2, 3, 4\right\}, \quad q = 1, 2, \ldots, N.$$

For any fixed first-stage feasible decision $\boldsymbol{u}$, when ζ_n takes on the value $\hat{\zeta}_n^q=(\hat{\zeta}_{n,1}^q, \hat{\zeta}_{n,2}^q, \hat{\zeta}_{n,3}^q, \hat{\zeta}_{n,4}^q)$, we solve the second-stage linear programming (4) by simplex algorithm, and denote the optimal value by $Q(\boldsymbol{u}, \hat{\zeta}_n^q)$, $q = 1, 2, \ldots, N$. The possibility that $Q(\mathbf{u}, \zeta_n)$ takes on the value $Q(\boldsymbol{u}, \hat{\zeta}_n^q)$ is $\upsilon^q = \mathrm{Pos}\{\zeta_n = \hat{\zeta}_n^q\}$.

Rearrange the superscript q of $Q(\boldsymbol{u}, \hat{\zeta}_n^q)$ such that

$$Q(\boldsymbol{u}, \hat{\zeta}_n^1) \le Q(\boldsymbol{u}, \hat{\zeta}_n^2) \le \cdots \le Q(\boldsymbol{u}, \hat{\zeta}_n^N).$$

Then the value of $\boldsymbol{E}_\xi[Q(\mathbf{u}, \zeta_n)]$ is calculated by

$$f_m(\boldsymbol{u}) = \sum_{q=1}^{N} w_q Q(\boldsymbol{u}, \hat{\zeta}_n^q), \tag{16}$$

where the weights $w_q, q = 1, 2, \ldots, N$, are given by

$$w_q = \frac{1}{2}\left(\max_{p=1}^{q} \upsilon^p - \max_{p=0}^{q-1} \upsilon^p\right) + \frac{1}{2}\left(\max_{p=q}^{N} \upsilon^p - \max_{p=q+1}^{N+1} \upsilon^p\right) \tag{17}$$

with $\upsilon^0 = \upsilon^{N+1} = 0$.

From the computational process of $\boldsymbol{E}_\xi[Q(\mathbf{u}, \zeta_n)]$, we have

$$Q(\boldsymbol{u}, \hat{\zeta}_n^1) - f_m(\boldsymbol{u}) \le Q(\boldsymbol{u}, \hat{\zeta}_n^2) - f_m(\boldsymbol{u}) \le \cdots \le Q(\boldsymbol{u}, \hat{\zeta}_n^3) - f_m(\boldsymbol{u}).$$

As a consequence, the value of semivariance $SV^+[Q(\mathbf{u}, \zeta_n)]$ is computed by

$$h_m(\boldsymbol{u}) = \sqrt{\sum_{q=n_{min}}^{N} w_q \left(Q(\boldsymbol{u}, \hat{\zeta}_n^q) - f_m(\boldsymbol{u})\right)^2}, \tag{18}$$

where n_{min} is the smallest index q such that $Q(\boldsymbol{u}, \hat{\zeta}_n^q) - f_m(\boldsymbol{u}) \ge 0$, and the weights $w_q, q = 1, 2, \ldots, N$, are given by Eq. (17).

Using the approximation method, we obtain the approximating optimization model of the SCND problem (9),

$$\begin{aligned} \min = &\sum_{j\in J} cm_j u_j + \sum_{l\in L} cw_l w_l + \sum_{q=1}^{N} w_q Q(\boldsymbol{u}, \hat{\zeta}_n^q) \\ &+ \sqrt{\sum_{q=n_{min}}^{N} w_q \left(Q(\boldsymbol{u}, \hat{\zeta}_n^q) - E\right)^2} \end{aligned} \tag{19}$$

subject to: $u_j, w_l \in \{0, 1\}, \quad \forall j \in J, \forall l \in L,$

where $E = \sum_{q=1}^{N} w_q Q(\boldsymbol{u}, \hat{\zeta}_n^q)$, and $Q(\mathbf{u}, \zeta_n(\gamma))$ is the optimal value of the following program model

$$\begin{aligned} Q(\mathbf{u}, \zeta_{\mathbf{n}}(\gamma)) = \min &\sum_{i\in I}\sum_{j\in J} \hat{cp}_{ij}^n(\gamma) x_{ij} \\ &+ \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} \hat{cp'}_{jlk}^n(\gamma) y_{jlk} \\ &+ \sum_{k\in K}\sum_{l\in L}\sum_{m\in M} \hat{cp''}_{lmk}^n(\gamma) z_{lmk} \\ &+ \sum_{i\in I} cq_i^I \sum_{j\in J} x_{ij} \\ &+ \sum_{k\in K}\sum_{j\in J} cq_{jk}^J \sum_{l\in L} y_{jlk} \\ &+ \sum_{l\in L}\sum_{m\in M} p_{mk}\tilde{z}_{mk} \\ \text{subject to: } &\sum_{l\in L} z_{lmk} + \tilde{z}_{mk} \ge \hat{d}_{mk}^n(\gamma), \quad \forall m \in M, \forall k \in K \\ &\sum_{i\in I} x_{ij} = \sum_{k\in K} r_k \sum_{l\in L} y_{jlk}, \quad \forall j \in J \\ &\sum_{j\in J} y_{jlk} = \sum_{m\in M} z_{lmk}, \quad \forall l \in L, \forall k \in K \\ &\sum_{j\in J} x_{ij} \le s_i, \quad \forall i \in I \\ &\sum_{k\in K} r_{jk}^P \sum_{l\in L} y_{jlk} \le a_j u_j, \quad \forall j \in J \\ &\sum_{k\in K} r_{lk}^L \sum_{j\in J} y_{jlk} \le h_l w_l, \quad \forall l \in L \\ &x_{ij} \ge 0, y_{jlk} \ge 0, z_{lmk} \ge 0, \quad \forall i, j, k, l, m. \end{aligned} \tag{20}$$

Problem (19) is a two-stage mixed-integer programming model. The following theorem shows the objective value of problem (19) converges to that of the SCND problem (9).

Theorem 2 *Consider two-stage mean-risk SCND problem (9). Suppose the parameter $\xi = (\xi_1, \xi_2, \xi_3, \xi_4)$ is a continuous fuzzy vector and its possibility distribution is given by Eq. (15). If the sequence $\{\zeta_n\}$ of fuzzy vectors is the discretization of fuzzy vector ξ, then for any fixed feasible decision $\mathbf{u} \in U$, we have*

$$\lim_{n\to\infty} \boldsymbol{E}_{\zeta_{\mathbf{n}}}[Q(\mathbf{u}, \zeta_n)] = \boldsymbol{E}_\xi[Q(\mathbf{u}, \xi)],$$

$$\lim_{n\to\infty} \sqrt{SV_{\zeta_n}^+[Q(\mathbf{u}, \zeta_n)]} = \sqrt{SV_\xi^+[Q(\mathbf{u}, \xi)]}.$$

Proof For any fixed feasible solution $\mathbf{u} \in U$ and every realization $\xi(\gamma)$ of fuzzy vector ξ, the second-stage value function $Q(\mathbf{u}, \xi)$ is continuous on the compact subset $\prod_{i=1}^4 [a_i, b_i]$ of Ξ, where Ξ is the support of ξ. Therefore, the suppositions of the theorem satisfy the conditions of (Liu 2006, Theorem 3), which verifies the assertions of the theorem.

Problem (19) is a nonlinear mixed-integer programming, it is in general a nonconvex programming model. In the next section, we will design a hybrid MA to solve this hard optimization problem.

Hybrid solution methods

The MA combines the global search and the local search by using the genetic algorithm to perform the exploration and exploitation, it keeps well the balance between the intensification and diversification.

Representation

In the implementation of the MA, each chromosome U represents a first-stage decision vector. The component of U includes 0s (zeros) and 1s (ones), where the value 1 indicates that the corresponding plant or warehouse is built, while the value 0 indicates that the corresponding plant or warehouse is not built. A chromosome is represented by [plant1, plant2, . . ., plantJ, warehouse1, . . ., warehouseL], where the dimension of the chromosome is the total number of plants and warehouses. For example, if J = 4 and L = 5, then the chromosome [0 1 0 1 0 0 1 1 0] means that plants 2, 4 and warehouses 3, 4 are built, while plants 1, 3 and warehouses 1, 2 and 5 are not built.

Initialization process

To generate a feasible chromosome, we first select randomly a plant from the set {plant1, . . . , plantJ} and set its value as 1, and select randomly a warehouse from the set {warehouse1, . . . , warehouseL} and set its value as 1. Then the values of other components are set randomly from the set {0, 1}. Repeating this process pop_size times, we obtain pop_size feasible chromosomes $U_1, U_2, \ldots, U_{pop\_size}$.

The fitness of each chromosome is given by the value of objective function. We compute the values of objective function by the approximation method, in which we employ LINGO 8.0 software to solve the second-stage linear programming problem.

Recombination processes

Let the parameter $P_r \in (0, 1)$ be the probability of recombination. We select the chromosomes as parents for recombination by the roulette wheel method. For any two parents A and B, we have two operations in their recombination processes. The first operation is the single-point recombination. Let the recombination point k in the chromosome be generated randomly from the interval [1, N], where $N = J + L$. Then the parents, A and B, generate their offsprings C and D as follows. Chromosome C inherits the first part of parent A and the second part of parent B, while chromosome D inherits the first part of parent B and the second part of parent A. Figure 1 illustrates this single-point recombination process.

The second operation is the section recombination. We generate randomly two different points from interval [2, N-1], and use the two points to divide the parents A and B into three sections respectively. Then we exchange the middle sections of parents A and B to generate their offsprings C and D respectively. Figure 2 illustrates this section recombination process.

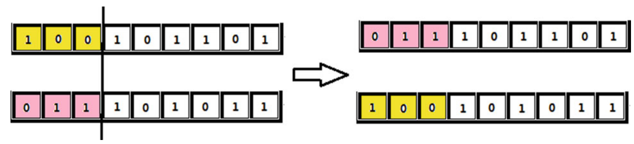

Fig. 1 An example of the single-point recombination operation

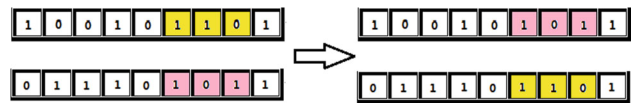

Fig. 2 An example of the section recombination operation

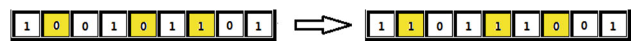

Fig. 3 An example of the three-point mutation operation

Fig. 4 An example of the mirror operation

Mutation processes

After recombining processes, we continue the mutation operation for the obtained chromosomes. A multi-point mutation operation is adopted to renew the chromosomes. Let the parameter $P_m \in (0, 1)$ be the probability of mutation. Then we select randomly k components from the chromosome $(1 \leq k \leq N/3)$, and change their values as follows. If the value in the selected component is 1, then we change it into 0; otherwise we change the value 0 into 1. In Fig. 3, we give a three-point mutation example for the case $k = 3$.

Local search

In our MA, the RVNS method is adopted as the local search procedure, and it contains the mirror and shift neighborhood structures. The chromosomes in the mirror neighborhood are generated as follows. We first generate randomly two different points from the interval [1, N] to produce a section between the two points. Then we generate a new chromosome by reversing the subsequence's arrangement of the produced section. If the new chromosome has the same arrangement with the original chromosome, we apply a one-point mutation operation to renew it. An example of the mirror operation is shown in Fig. 4.

The chromosomes in the shift neighborhood are generated as follows. We first generate randomly two different points from the interval [1, N] to produce a section between the two points. Then we generate a new chromosome by removing the produced section from its current position and inserting

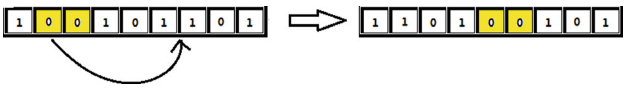

Fig. 5 An example of the shift operation

it in another randomly selected position. Figure 5 gives an example to illustrate this shift operation.

The RVNS method starts from the original chromosome and finds a better solution (chromosome) in the mirror and shift neighborhoods. Firstly, the RVNS finds a better solution in the mirror neighborhood, then it finds a better solution in the shift neighborhood. If a better chromosome is found in one neighborhood, then we update the original chromosome using the better solution and continue the search procedure. Otherwise, the RVNS changes its searching neighborhood and finds a better solution in another new neighborhood. The stop condition of our RVNS method is the maximum searching time. The simplicity of the local search process is explained by the following VC++ pseudocode.

```
static void LocalSearchRVNS(int Chrom[N0+1])
{   int i,j,t;
    int temp[N0+1];
    Copy(Chrom,temp);t=1;
    for(i=1;i<3;i++)
    {   if(i==1)
           Mirror(temp);
        if(i==2)
           Shift(temp);
        if(Fitness(temp)<Fitness(Chrom))
        {
                Copy(temp,Chrom);i=1;
           }
           if(t>T_max)
              break;
           t++;
    }
}
```

Finally, we summarize the process of the hybrid MA in Algorithm 1, and provide the flowchart of the hybrid MA in Fig. 6.

Algorithm 1 Hybrid Memetic algorithm for SCND

Set Gen, Gbest.
Initialization();
LocalSearchRVNS();
for $i = 1; i < Gen; i++$ **do**
  Selection();
  Recombination();
  Mutation();
  LocalSearchRVNS();
end for
return Gbest;

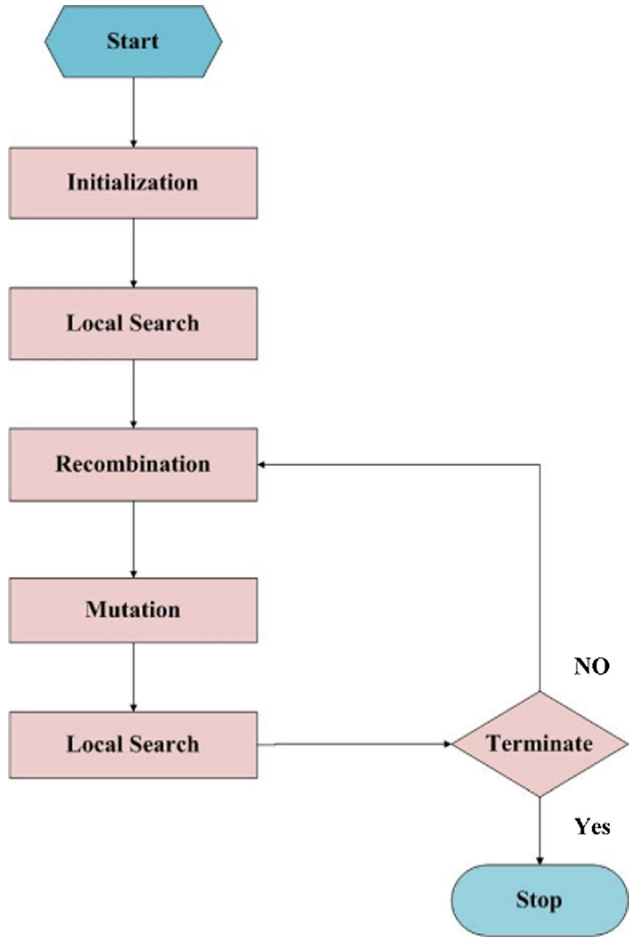

Fig. 6 The flowchart of the hybrid MA

Numerical experiments and comparison study

In this section, we present the computational results of numerical experiments to assess the behavior of the designed solution method. The hybrid MA has been coded in C++. When computing the values of objective function, we employ the LINGO 8.0 software to solve the second-stage linear programming models. The numerical experiments are executed on a personal computer (Intel Pentium(R) Dual-Core E5700 3.00GHZ CPU and RAM 2.00GB), using the Microsoft Windows 7 operating system.

Application examples and computational results

In this section, we consider a supply, production and distribution network design problem. In our application example, there are 4 raw material suppliers, 5 plants, 6 warehouses, 13 customers and 2 types of products. Suppose the supplers, plants, warehouses and customers locate in 28 cities of China, and their supply chain network is plotted in Fig. 7.

Fig. 7 The network of the application example

Table 2 The distance (km) from supplier s_i to plant p_j

	p_1	p_2	p_3	p_4	p_5
s_1	170	472	651	493	1,023
s_2	440	714	655	370	1,146
s_3	1,103	757	270	424	480
s_4	1,040	957	421	365	918

Table 3 The distance (km) from plant p_j to warehouse w_l

	p_1	p_2	p_3	p_4	p_5
w_1	360	290	460	415	783
w_2	267	344	1,051	778	1,000
w_3	1,034	1,285	972	683	1,542
w_4	1,402	1,300	723	715	1,123
w_5	1,223	941	387	624	624
w_6	1,016	1,138	762	491	1,313

In this problem, we assume the transportation costs among cities are related to the distances and the fuel prices, and the fuel prices are uncertain with time varying. The transportation cost $cp_{ij}(\gamma)$ is a linear function of $\xi_1, cp_{ij}(\gamma) = cp\_a_{ij} \times \xi_1(\gamma) + cp\_b_{ij}$, where ξ_1 is the unit fuel cost per kilometer from suppliers to plants, cp_a_{ij} is the distance between two cities and cp_b_{ij} is the extra cost due to the influence of other factors. Similarly, the transportation cost $cp'_{jlk}(\gamma)$ is a linear function of $\xi_2, cp'_{jlk}(\gamma) = cp'\_r_k \times cp'\_a_{jl} \times \xi_2(\gamma) + cp'\_b_{jlk}$, where ξ_2 is the unit fuel cost per kilometer from plants to warehouses, cp'_a_{jl} is the distance between two cities, cp'_r_k is the cost coefficient for product k and cp'_b_{jlk} is the extra cost; the transportation cost $cp''_{lmk}(\gamma)$ is a linear function of $\xi_3, cp''_{lmk}(\gamma) = cp''\_r_k \times cp''\_a_{lm} \times \xi_3(\gamma) + cp''\_b_{lmk}$, where ξ_3 is the unit fuel cost per kilometer from warehouses to customers, cp''_a_{lm} is the distance between two cities, cp''_r_k is the cost coefficient for product k and cp''_b_{lmk} is the extra cost; the demand $d_{mk}(\gamma)$ is a linear function of $\xi_4, d_{mk}(\gamma) = d\_r_k \times d\_a_m \times \xi_4(\gamma) + d\_b_{mk}$, where ξ_4 is the unit demand per million people, d_a_m is the population size of city m, d_r_k is the demand coefficient for product k and d_b_{mk} is the extra demand.

Table 4 The distance (km) from warehouse w_l to customer c_m

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}
w_1	606	574	547	674	722	1,106	366	517	293	829	1,013	1,083	1,131
w_2	253	273	381	390	1,095	1,455	250	631	587	989	1,248	1,398	1,495
w_3	1,510	1,533	1,073	1,326	881	484	1,365	1,446	1,138	1,650	1,637	1,547	1,172
w_4	884	813	1,002	1,102	568	1,162	538	272	163	388	542	660	873
w_5	1,450	1,375	1,404	1,564	258	846	1,086	828	648	782	547	407	282
w_6	1,463	1,457	1,076	1,333	665	334	1,260	1,286	1,462	1,455	1,423	1,318	959

Table 5 The population (million) of the customer c_m

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}
Population	18.8	8.5	1.9	2.0	3.0	4.3	4.5	7.2	7.0	22.3	3.0	3.5	12.7

For the required data in our application example, the distances from the raw material supplier cities to the plant cities are given in Table 2; the distances from the plant cities to the warehouse cities are collected in Table 3; the distances from the warehouse cities to the customer cities are provided in Table 4, and the populations of 13 cities are given in Table 5. In addition, the extra costs cp_b_{ij}, cp'_b_{jlk} and cp''_b_{lmk} are generated randomly from the interval [1, 4]; the extra demand d_b_{mk} is generated randomly from the interval [10, 30]; the cost coefficients cp'_r_k, cp''_r_k and demand coefficient d_r_k are generated randomly from the interval [0.7, 3]; the building costs cm_j and cw_l are generated randomly from the interval [30000, 50000]; the specific capabilities a_j, h_l and s_i are generated randomly from the interval [1000, 6000]; the requirement parameters r^P_{jk}, r^L_{lk} and r_k are generated randomly from the interval [0.5, 3], and the unit production costs cq^J_{jk} and cq^I_i are generated randomly from [10, 30]. In our application example, we assume the fuzzy vector $(\xi_1, \xi_2, \xi_3, \xi_4)$ has the following joint possibility distribution

$$\pi(x) = \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma(x-\mu)\right\},$$

where $\Sigma = U^T U$, and U is a nonsingular upper triangular matrix. The elements of U are generated randomly from the interval [0.1, 0.3], and the value of $\mu = (\mu_1, \mu_2, \mu_3, \mu_4)^T$ is generated randomly from the interval [15, 35].

For $\lambda = 10$, we employ the hybrid MA to solve the instance by using various sizes of sample points. During the numerical experiments, we increase the number of realizations of fuzzy variable ξ_i from 3 to 12, $i = 1, 2, 3, 4$. As a consequence, the total number of realizations of fuzzy vector $(\xi_1, \xi_2, \xi_3, \xi_4)$ increases from 3^4 to 12^4. Figure 8 illustrates the convergence of the mean value and standard semivariance in our numerical experiments.

Comparison study

In our SCND problem, the MRVFS measures the value of knowing and using the possibility distributions on future realizations of fuzzy parameters, so the quantity MRVFS is relevant to our mean-risk SCND problem. We verify the assertion via numerical experiments, and report the computational results of the MREV, MRRP and MRVFS in Table 6.

The risk coefficient $\lambda = 0$ corresponds to the two-stage risk-neutral SCND problem. In this case, the network structure of our example is [1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0]. On the other hand, the risk coefficient $\lambda > 0$ corresponds to the two-stage mean-risk SCND problem. In this case, we can obtain different network structures by using various values of the risk coefficient. For instance, when $\lambda = 10$, the network structure is [1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0]; when $\lambda = 10000$, the network structure is [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1].

Furthermore, from Table 6, we can see that the larger the value of the risk coefficient λ, the larger the value of the MRVFS. That is, we obtain large benefits from solving the two-stage mean-risk SCND problem.

Conclusions

In this paper, we studied the SCND problem under uncertainty, where the customer demands and transportation costs are characterized by fuzzy variables with known possibility distributions. The major new results include the following several aspects.

(i) A new two-stage mean-risk SCND problem was proposed, which incorporates the trade-off between the expected costs and the risk brought out by fuzzy costs. We specified the standard semivariance as a novel mea-

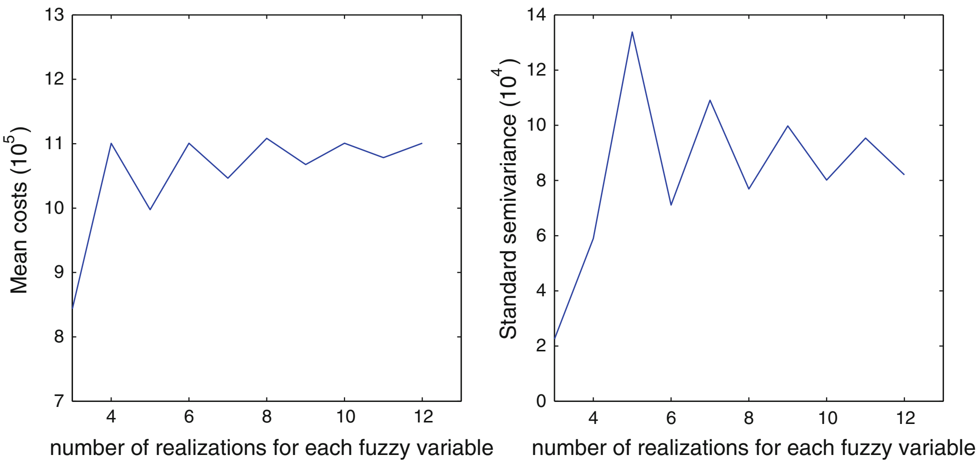

Fig. 8 The convergence of the mean value and standard semivariance with λ = 10

Table 6 The value of fuzzy solution for application example

λ	MRRP	MREV	MRVFS
0	1095763	1115969	20206
1	1156556	1236162	79606
10	1690238	2317902	627664
100	6997587	13135301	6137714
1000	59483417	121309291	61825874
10000	584274329	1203049192	618774863

sure for gauging the risk brought out by asymmetric fuzzy costs.

(ii) We introduced the concepts of the MRRP, MREV and MRVFS for the proposed two-stage mean-risk SCND problem. Our emphasis in this paper is on the SCND problem where no further information about costs and demands is available, therefore the quantity MRVFS is practically relevant to our mean-risk SCND problem.

(iii) When the customer demands and transportation costs have continuous possibility distributions, we approximated the continuous fuzzy vector by a sequence of discrete fuzzy vectors. To ensure the solution quality, we discussed the convergence of the approximation method in Theorem 2.

(iv) To solve the proposed SCND problem, we designed a hybrid MA, in which the mirror operator and the shift operator in the RVNS were used to act as the local search. We also provided an application example with 4 raw material suppliers, 5 plants, 6 warehouses, 13 customers and 2 types of products, and solved the instance by the designed hybrid MA. The computational results demonstrated the effectiveness of the developed solution method.

Acknowledgments The authors wish to thank Editors and anonymous referees, whose valuable comments led to an improved version of this paper. This work was supported by the National Natural Science Foundation of China, the Natural Science Foundation of Hebei Province (A2011201007), and the Training Foundation of Hebei Province Talent Engineering.

References

Aikens, C. H. (1985). Facility location models for distribution planning. *European Journal of Operational Research*, *22*(3), 263–279.

Alonso-Ayuso, A., Escudero, L. F., Garin, A., Ortuno, M. T., & Perez, G. (2003). An approach for strategic supply chain planning under uncertainty based on stochastic 0–1 programming. *Journal of Global Optimization*, *26*, 97–124.

Bachlaus, M., Pandey, M. K., Mahajan, C., Shankar, R., & Tiwari, M. K. (2008). Designing an integrated multi-echelon agile supply chain network: A hybrid taguchi-particle swarm optimization approach. *Journal of Intelligent Manufacturing*, *19*(6), 747–761.

Bidhandi, H. M., Yusuff, R. M., Ahmad, M. M. H. M., & Bakar, M. R. A. (2009). Development of a new approach for deterministic supply chain network design. *European Journal of Operational Research*, *198*(1), 121–128.

Cheung, R. K. M., & Powell, W. B. (1996). Models and algorithms for distribution problems with uncertain demands. *Transportation Science*, *30*(1), 43–59.

Cho, D. W., & Lee, Y. H. (2012). Bullwhip effect measure in a seasonal supply chain. *Journal of Intelligent Manufacturing*, *23*(6), 2295–2305.

Cohen, M. A., & Lee, H. L. (1989). Resource deployment analysis of global manufacturing and distribution networks. *Journal of Manufacturing and Operations Management*, *2*, 81–104.

Geoffrion, A. M., & Graves, G. W. (1974). Multi-commodity distribution system design by Benders decomposition. *Management Science*, *20*(5), 822–844.

Geoffrion, A. M., & Powers, R. F. (1995). Twenty years of strategic distribution system design: An evolutionary perspective. *Interfaces*, *25*(5), 105–127.

Georgiadis, M. C., Tsiakis, P., Longinidis, P., & Sofioglou, M. K. (2011). Optimal design of supply chain networks under uncertain transient demand variations. *Omega*, *39*(3), 254–272.

Giannoccaro, I., Pontrandolfo, P., & Scozzi, B. (2003). A fuzzy echelon approach for inventory management in supply chains. *European Journal of Operational Research*, *149*(1), 185–196.

Jolai, F., Amalnick, M. S., Alinaghian, M., Shakhsi-Niaei, M., & Omrani, H. (2011). A hybrid memetic algorithm for maximizing the weighted number of just-in-time jobs on unrelated parallel machines. *Journal of Intelligent Manufacturing*, *22*(2), 247–261.

Kubat, C., & Yuce, B. (2012). A hybrid intelligent approach for supply chain management system. *Journal of Intelligent Manufacturing*, *23*(4), 1237–1244.

Lee, H. Y., Kang, A. H. I., & Yang, C. Y. (2012). A fuzzy ANP model for supplier selection as applied to IC packaging. *Journal of Intelligent Manufacturing*, *23*(5), 1477–1488.

Liu, B., & Liu, Y. (2002). Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems*, *10*(4), 445–450.

Liu, Y. (2005). Fuzzy programming with recourse. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *13*(4), 381–413.

Liu, Y. (2006). Convergent results about the use of fuzzy simulation in fuzzy optimization problems. *IEEE Transactions on Fuzzy Systems*, *14*(2), 295–304.

Mizgiera, K. J., Wagnera, S. M., & Holystb, J. A. (2012). Modeling defaults of companies in multi-stage supply chain networks. *International Journal of Production Economics*, *135*(1), 14–23.

Petrovic, D., Roy, R., & Petrovic, R. (1999). Supply chain modelling using fuzzy sets. *International Journal of Production Economics*, *59*(1–3), 443–453.

Pishvaee, M. S., Farahani, R. Z., & Dullaret, W. (2010). A memetic algorithm for bi-objective integrated forward/reverse loistics network design. *Computers & Operations Research*, *37*(6), 1100–1112.

Sabri, E. H., & Beamon, B. M. (2000). A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega*, *28*(5), 581–598.

Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, *167*(1), 96–115.

Sun, G., Liu, Y., & Lan, Y. (2011). Fuzzy two-stage material procurement planning problem. *Journal of Intelligent Manufacturing*, *22*(2), 319–331.

Urselmann, M., Barkmann, S., Sand, G., & Engell, S. (2011). A memetic algorithm for global optimization in chemical process synthesis problems. *IEEE Transanctions on Evolutionary Computation*, *15*(5), 659–683.

Wang, J., & Shu, Y. (2005). Fuzzy decision modeling for supply chain management. *Fuzzy Sets and Systems*, *150*(1), 107–127.

Wong, J. T. (2012). DSS for 3PL provider selection in global supply chain: Combining the multi-objective optimization model with experts' opinions. *Journal of Intelligent Manufacturing*, *23*(3), 599–614.

Yang, G., Liu, Y., & Yang, K. (2011). Modeling supply chain network design problem with joint service level constraint. *Advances in Intelligent and Soft Computing*, *123*, 311–318.

Yeh, W. C. (2006). An efficient memetic algorithm for the multi-stage supply chain network problem. *The International Journal of Advanced Manufacturing Technology*, *29*(7–8), 803–813.

Zhou, C., Zhao, R., & Tang, W. (2008). Two-echelon supply chain games in a fuzzy environment. *Computers & Industrial Engineering*, *55*(2), 390–405.