International Journal of Innovative
Computing, Information and Control
Volume 10, Number 4, August 2014

ICIC International ©2014 ISSN 1349-4198
pp. 1495–1508

A TWO-STAGE PARTICLE SWARM OPTIMIZATION FOR VIRTUAL ENTERPRISE RISK MANAGEMENT

Zhizhong Ding[1], Fuqiang Lu[2] and Hualing Bi[2]

[1]Textile Division
Tianjin Polytechnic University
No. 63, Chenglin Road, Hedong District, Tianjin 300387, P. R. China
dzz2000@sina.com

[2]School of Economics and Business
Northeastern University at Qinhuangdao
No. 143, Taishan Road, Haigang District, Qinhuangdao 066004, P. R. China
fqlu@neuq.edu.cn; bihualing081@126.com

Received July 2013; revised December 2013

Abstract. *In this paper, we propose a novel Two-stage Particle Swarm Optimization (TSPSO) to solve the problem of virtual enterprise (VE) risk management. A two-level risk management management model is considered. In the top level, the objective of the owner is to maximum the benefit of risk management for the whole VE. In the base level, the partners aim to maximum their benefit of risk management. The captured problem is very challenging due to its hierarchical structure and its time complexity, so the TSPSO is designed for the risk management problem. The TSPSO has two searching processes, namely, "top-search", the searching process for the top level, and "base-search", the searching process for the base level. The performance of TSPSO is then compared with both the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), in terms of efficient frontiers, fitness values, convergence rates, computational time consumption and reliability. The experimental results show that TSPSO is more efficient and reliable for the two-level risk management problem than the other tested methods. The TSPSO is an improvement on the previous single searching process and single swarm optimizers, which has great potential to be used to solve the optimization problems with hierarchical structure and high time complexity.*
Keywords: Virtual enterprise, Risk management, Genetic algorithm, Particle swarm optimization, Two-level particle swarm optimization

1. **Introduction.** Global competition and the varieties of the customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. It requires these enterprises quick response, lower costs, and greater customization. Virtual Enterprise (VE) is a joint venture designed to be a temporary alliance of member companies. These enterprises combine to take advantage of the market opportunities to develop and produce products for fulfilling consumer requirements in the rapidly changing environment of the global manufacturing area. However, in the paradigm of the VE, there are various sources of risks that may threaten the security of the VE, such as market risk, credit risk, operational risk, liquidity risk and others [1, 2]. Risk measurement and management of a VE have received considerable interest among researchers and managers of enterprises. Various models and algorithms are developed to provide a more scientific and effective way for managing the risk of a VE. In the risk management of VE, The risk control strategy selection is a combinatorial optimization problem, which is usually a NP-hard problem while the size of problem increase. Some optimization techniques are found to be useful in enhancing the profitability from trading

these instruments. For example, some standard optimization techniques can be used to maximum the benefit of risk management. Ip et al. (2003) [3] consider minimizing the risk in selecting partners and ensuring the due date of a project in a VE. They propose a risk-based partner selection model. By exploring the characteristics of the problem considered and the knowledge of project scheduling, a Rule-based Genetic Algorithm (R-GA) with embedded project scheduling is developed to solve the problem. Huang et al. (2008) [4] focus on two main features of the VE, project mode and uncertain factors. They establish the fuzzy synthetic evaluation embedded nonlinear integer programming model of risk programming for the VE and present a tabu search (TS) algorithm with an embedded fuzzy synthetic evaluation for the model. Tao et al. [5] introduces a novel quantum multi-agent evolutionary algorithm (QMAEA) for addressing partner selection problems in a virtual enterprise. In QMAEA, each agent represented by a quantum-bit is defined as a candidate solution, and agents can reproduce, perish, compete for survival, observe and communicate with the environment. Operators such as energy evaluation, competition, crossover, mutation, and trimming are designed to specify the evolvement of QMAEA. Three evolutionary strategies are designed to balance the exploration and exploitation of QMAEA. The effectiveness and scalability of the proposed QMAEA in addressing PSP is demonstrated with experimental results and comparisons. Huang et al. [6] developed a Distributed Decision Making (DDM) model for risk management of the VE. The model has two levels, namely, the top model and the base model, which describe the decision processes of the owner and the partners of the VE respectively. A simple information situation, symmetric information, between owner and partners is considered. A Particle Swarm Optimization (PSO) algorithm is then designed to solve the resulting optimization problem. The result shows that the proposed algorithm is effective and the two-level model can help improve the description of the relationship between the owner and the partners, which is helpful to reduce the risk of the VE. Shao et al. (2012) [7] develops a novel multiswarm particle swarm optimizer called PS^2O to solve the optimization model for risk management in a virtual enterprise. The main idea of PS^2O is to extend the single population PSO to the interacting multiswarm model by constructing hierarchical interaction topology and enhanced dynamical update equations. With five mathematical benchmark functions, PS^2O is proved to have considerable potential for solving complex optimization problems. Huang et al. (2013) [8] considered the coordination under symmetric information between owner and partners in risk management of VE. A centralized mechanism is given as the base case, and then a distributed decision-making (DDM) mechanism with incentive scheme is introduced to establish a practicable strategic partnership. A Particle Swarm Optimization (PSO) algorithm is then designed to solve the resulting optimization problem. The study shows that the DDM mechanism with incentive scheme can improve the overall benefit of risk management beyond the centralized one.

In this paper, a two-level model with stochastic variables is introduced. And then a two-stage particle swarm optimization (TSPSO) is designed to provide an effective way to solve the problem. TSPSO has some characteristics of the particle swarm optimization (PSO) algorithm, such as collectiveness and mutual learning among individuals. However, unlike PSO, TSPSO is a global convergent algorithm and has stronger search ability than PSO. The efficiency and effectiveness of TSPSO in the two-level optimization problem solving, PSO and genetic algorithm (GA) are also tested for the purpose of performance comparison.

The remainder of this paper is structured as follows. Section 2 presents the two-level risk management model. In Section 3, the TSPSO is given. Numerical examples and insights are depicted in Section 4. Concluding remarks are then given in Section 5.

2. **Two-Level Risk Management Model.** The risk management problem can be described as follows.

Assume an enterprise/owner find a business project consisting of several sub-projects. The owner is not able to complete the whole project using its own capacity. Therefore, it has to invite partners for the sub-projects. Of course, the risk management of the sub-projects is also an important responsibility of partners. The owner determines the upper bound of the budget for each sub-project first. The partners who accept the budget condition will respond to the risk management of its sub-project and propose the benefit of risk management they need to finish according to the resources they have. The benefit of risk management of sub-project should beyond the target value which is required by the owner. In this process, the owner manages the risk of the entire project to ensure the success of VE, while maximize the benefit of risk management from the entire project.

For each partner, there are several risk factors which threaten its safety. Being dealt with risk control strategies, the risk factor will be controlled by some way. There are some strategies for each risk factor. The effects of strategies on the corresponding risk are different. Thus, the description for each risk factor and the cost of the different control strategies are different. The partner has to maximize its benefit of risk management by optimally combining these strategies, under the budget allocated by the owner. The benefit of risk management for each partner/subproject is a difference between the initial risk loss of the subproject and the risk loss under risk management. In the decision process, there are private information between the owner and partners, which is the major source of uncertainty. For the owner, the accurate information of partners is difficult to acquire, such as the probability of risk occurrences for each risk factor, the risk loss for each risk factor, the strategies selected for risk factors.

2.1. **Notations.** The following notations and assumptions are used in two-level model.

Notations:

$L_0^{initial}$	initial state of risk loss for owner ($).
$L_i^{initial}$	initial state of risk loss for partner i in the VE ($), $i = 1, 2, \ldots, M$.
M	number of partners in the VE.
x_0	budget for owner.
$L_0(x_0)$	risk loss of owner under budget x_0 ($).
x_i	budget for Partner i, $i = 1, 2, \ldots, M$.
x_i^*	the real budget for Partner i, $i = 1, 2, \ldots, M$.
$\hat{Benefit}_i(x_i)$	anticipated benefit of partner i from risk management under budget x_i ($).
$Benefit_i(x_i)$	benefit of partner i from risk management.
$L_i(x_i)$	risk loss of Partner i under budget x_i.
B_{max}	upper bound of the total budget for risk management of the VE ($).
TB_i	target benefit of risk management from partner i, required by the owner ($).
B_i	upper bound of the budget for partner i ($), $i = 1, 2, \ldots, M$.
$\hat{y}_{ij}$	anticipated risk control strategy selected for risk factor j of partner i.
y_{ij}	risk control strategy selected for risk factor j of partner i.
$l_{ij}(y_{ij})$	risk loss from risk factor j of partner i under strategy y_{ij}.
$\hat{p}_{ij}$	anticipated probability of risk occurrence for risk factor j of partner i.
p_{ij}	probability of risk occurrence for risk factor j of partner i.
α	confidence level of the chance constraint programming model.
$C_{ij}(y_{ij})$	cost of Partner i under the risk control action for the risk factor j ($).
W_{ij}	available strategy number for risk factor j of partner i.
N_i	number of risk factor for partner i.

2.2. **The two-level risk management model.** In the process of risk management, firstly the owner allocates budget among members of VE within the given total budget to maximize the benefit of risk management of VE. Then the partners select optimal risk control strategies to maximize its benefit of risk management under the allocated budget. Since the asymmetric information, if the owner wants to get a better benefit, the owner has to anticipate the situation of the partners before making decision. The information is asymmetry between owner and partners, the owner can not anticipate the accurate situation of partners. The decision process of risk management is described by a two-level model, which is demonstrated below.

The decision process of owner and partners are described by the top level and the base level respectively. There are two models in the top level, top model and anticipated base model.

The top level:

Top model:

$$\max[L_0^{initial} - L_0(x_0) - x_0] + \sum_{i=1}^{M} Bene\hat{f}it_i(x_i) \tag{1}$$

s.t.

$$\sum_{i=0}^{M} x_i \leq B_{\max} \tag{2}$$

$$Bene\hat{f}it_i(x_i) > TB_i, \quad i = 1, 2, \ldots, M. \tag{3}$$

$$x_i \in [0, B_i] \quad i = 0, 1, \cdots, M. \tag{4}$$

In top model, the decision maker is the owner who allocates the budget to each member of the VE, including itself. The owner's aim is to maximum the benefit of risk management for the whole VE, the decision variables are therefore given by $\{x_0, x_1, \ldots, x_M\}$. In order to get a better benefit, the benefit of risk management from partners should beyond a given level (TB_i) required by the owner, see Equation (3). Equation (2) and Equation (4) present the constraint of investment budget and the interval of decision variables respectively.

Anticipated base model i:

$$\max Bene\hat{f}it_i(x_i) = Benefit_i \tag{5}$$

s.t.

$$\Pr\left\{L_i^{initial} - \sum_{j=1}^{N_i} \hat{p}_{ij} l_{ij}(y_{ij}) - x_i \geq Benefit_i\right\} \geq \alpha \tag{6}$$

$$\sum_{j=1}^{N_i} C_{ij}(y_{ij}) \leq x_i \tag{7}$$

$$y_{ij} \in \{0, 1, \ldots, W_{ij}\} \quad j = 1, \cdots, N_i. \tag{8}$$

The owner has to anticipates M partners' decision process under the asymmetric information. So there are M anticipated base models. For the anticipated base model i, it is assumed that the probability of risk occurrence p_{ij} is an uncertainty factors for owner and it is taken as random variables. In other words, the owner only knows the distribution function of the probability $\Phi(\hat{p}_{ij})$, but not the accuracy condition. The risk loss of partner i is an expected value $p_{ij}l_{ij}(y_{ij})$, $i = 1, 2, \ldots, M$. The anticipated base model is a chance constraint programming model, α is the confidence level, and $\alpha \in (0, 1)$. The cost of risk management for partner i should no more than the allocated budget, see Equation (7). Equation (8) present the set of the decision variables y_{ij}.

The base level i:

$$\max Benefit_i(x_i^*) = L_i^{initial} - \sum_{j=1}^{N_i} p_{ij} l_{ij}(y_{ij}) - x_i^* \tag{9}$$

s.t.

$$\sum_{j=1}^{N_i} C_{ij}(y_{ij}) \le x_i^* \tag{10}$$

$$y_{ij} \in \{0, 1, \dots, W_{ij}\} \quad j = 1, \cdots, N_i. \tag{11}$$

In the base level, the decision makers are the partners. There are M base models or partners in the base level. Take the ith base model as an example, i.e., the model for Partner i. Partner i selects the optimal risk control strategies y_{ij} to maximize its benefit of risk management $Benefit_i$ under the allocated budget x_i^*. The cost of the risk management cannot exceed the allocated budget x_i^*, see Equation (10). Equation (11) presents the set of decision variables y_{ij}.

3. **The Two-Stage Particle Swarm Optimization.**

The two-level optimization problem presented in Section 2 is a nonlinear and integer programming problem with two-layer structures for which efficient algorithms do not exist [6, 7, 8, 9, 10]. In addition, the huge size of the problem will result in a high computational complexity when the number of credit entities and the number of consulting firms increase. Thus, considering the hierarchical structure of the problem, TSPSO is proposed which combine the "top-search" and "base-search" corresponding to the top model and base model of the problem receptively.

The original PSO was proposed by Kennedy and Eberhart in 1995 [11, 12, 13]. The PSO belongs to the category of swarm intelligence methods; it is also an evolutionary computation method inspired by social behavior observable in nature, such as flocks of birds and schools of fish. This study follows the "global best neighborhood topology" described by Kennedy et al., according to which, each particle remembers its best previous position and the best previous position visited by any particle in the whole swarm. In other words, a particle moves towards its best previous position and towards the best particle. Since PSO was first introduced to optimize various continuous nonlinear functions by Kennedy and Eberhard, it has been successfully applied to a wide range of applications including credit portfolio management. Recently, the PSO was improved to solve the problem with multi-level structures [14, 15, 16].

Figure 1 is the schematic presentation of the TSPSO. In the TSPSO, there are $N_{PT}+1$ populations, one top-population (top-pop) and N_{PB} base-populations (base-pop). Here N_{PT} is the population size of top-search. The top-pop is used to find the best solution of the whole problem. The N_{PB} base-pops are assigned to find the best solutions of the base model. Each base-pop corresponds to a specific particle of top-pop. The relationship between particles in top-pop and base-pop is shown in Figure 1. For the TSPSO, the search process starts from top-search. Each particle in the top-pop effects a base-pop by its specific allocated budget. Base-pops receive information from the particles of top-pop, their tasks are to find the best combination of risk control strategies. The particles in different base-pops cooperate with each other through the top-pop. The particles in the top-pop and base-pops can evolve by original PSO or its variants.

In the following subsections, the main steps of the TSPSO are described.

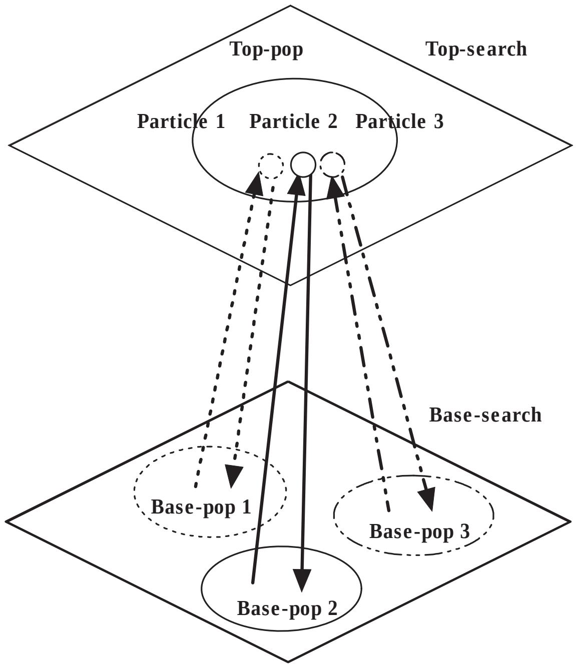

FIGURE 1. The schematic presentation of TSPSO

3.1. **Particle representation scheme.** First, the particle representation scheme for the TSPSO is presented. For the top-pop, the real number strings are selected as particles. Let N_{PT} denote the population size of the top-pop. For each $k = 1, 2, \ldots, N_{PT}$, define the k^{th} particle, $\mathbf{x}_k = (x_{k0}, \ldots, x_{kM})$, where $x_{ki} \in [0, 1]$. A particle represents a combination of allocated budget to the members of VE.

For the base-pops, the integer number strings are selected as particles. Let N_{PB} denote the population size of each base-pop. Then, for each $t = 1, 2, \ldots, N_{PB}$, the representation scheme of the t^{th} particle in the base-pop is $\mathbf{y}_t = (y_{t1}, y_{t2}, \ldots, y_{tN_i})$, where $y_{ti} \in \{1, 2, \ldots, W_i\}$. A particle represents a combination of risk control strategies for a partner.

3.2. **Initialization.** The initial top-pop with N_{PT} particles is generated first. Then, the initial base-pops with N_{PB} particles for each particle in the top-pop is generated. For the top-pop, random real numbers x_{ki} in the range $[0, B_i]$ are generated as the k^{th} particle $\mathbf{x}_k$, $i = 0, \ldots, M$. Random real numbers in the range $[-B_i, B_i]$ are generate as the initial velocity of the k^{th} particle v_{ki}.

For each base-pop, random integer numbers y_{ti} in the set $[1, \ldots, W_i]$ are generated as the t^{th} particle $\mathbf{y}_t$, $i = 1, \ldots, N_i$. Random integer numbers in the set $\{-W, \ldots, W\}$ are generated as the initial velocity of the t^{th} particle v_{ti}.

3.3. **Fitness function.** The fitness of the particles are calculated by the following fitness function:

For the top model,

$$\begin{aligned} F_T = {} & \left[L_0^{initial} - L_0(x_{k0}) - x_{k0}\right] + \sum_{i=1}^{M} Ben\hat{e}fit_i(x_{ki}) \\ & - \eta\left(\sum_{i=0}^{M} x_{ki} - B_{\max}\right)^+ - \lambda \sum_{i=0}^{M}\left(TB_i - Ben\hat{e}fit_i(x_{ki})\right)^+ \end{aligned} \tag{12}$$

Here F_T is the fitness of the particle $\mathbf{X}_k$. η and λ are punishment coefficients. Here notation $(\cdot)^+$ is defined as below,

$$(x)^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

For the chance constraint programming models (anticipated base models), Monte Carlo Simulation (MCS) is used to calculate the fitness under the chance constraint (2).

Step 1: Set Q' as integer part of αQ, Q is a integer number.

Step 2: Generate samples $p_{ij}^1, p_{ij}^2, \ldots, p_{ij}^Q$ of the random variable $\hat{p}_{ij}$ according to its distribution function $\Phi(\hat{p}_{ij})$, $i = 1$, $j = 1, 2, \ldots, N$.

Step 3: Using the samples to calculate the fitness by fitness function below,

$$F_{AB} = L_i^{initial} - \sum_{j=1}^{N_i} \hat{p_{ij}} l_{ij}(y_{tj}) - x_{ki} - \gamma \left(\sum_{l=1}^{N} C_{ij}(y_{tj}) - x_{ki} \right)^+ \tag{13}$$

For the Q samples, there are Q anticipated fitness, $F_{AB}^1, F_{AB}^2, \ldots, F_{AB}^Q$.

Step 4: Based on the law of large numbers, the value of $Benefit_i$ can be taken as the Q'th largest element in $\{F_{AB}^1, F_{AB}^2, \ldots, F_{AB}^Q\}$.

For the base models,

$$F_B = L_i^{initial} - \sum_{j=1}^{N_i} p_{ij} l_{ij}(y_{tj}) - x_{ki}^* - \gamma \left(\sum_{l=1}^{N} C_{ij}(y_{tj}) - x_{ki}^* \right)^+ \tag{14}$$

Here F_B is the fitness of particle $\mathbf{y}_t$; γ is a punishment coefficient.

3.4. **Updating of particles.** In this subsection, the formulas for updating the particles in top-pop and base-pop are presented respectively.

For the particles in top-pop,

$$v_{ki} = r v_{ki} + c_1 r_1 (p_{ki} - x_{ki}) + c_2 r_2 (p_{gi} - x_{ki}) \tag{15}$$

$$x_{ki} = x_{ki} + v_{ki} \tag{16}$$

Here, v_{ki} is the velocity of the k^{th} particle on the i^{th} bit, $v_{ki} \in [-x_i, x_i]$; x_{ki} represents the position of the k^{th} particle on the i^{th} bit; p_{ki} is the best position of the k^{th} particle on the i^{th} bit; p_{gi} is the best position of the population on the i^{th} bit so far; r is the inertia factor, $r \in [0, 1]$. r_1 and r_2 are random numbers, r_1, $r_2 \in [0, 1]$; c_1 and c_2 represent learning factors, where $c_1, c_2 \geq 0$; $k = 1, 2, \ldots, N_{PT}$ and $i = 1, 2, \ldots, M$.

For the particles in base-pops, the velocity and position are updated by (17) and (18) respectively,

$$v_{tj} = r v_{tj} + c_1 r_1 (p_{tj} - y_{tj}) + c_2 r_2 (p_{gj} - y_{tj}) \tag{17}$$

$$y_{tj} = \begin{cases} 1 & \text{if } v_{tj} + y_{tj} \leq 1 \\ v_{tj} + y_{tj} & \text{if } 1 < v_{tj} + y_{tj} \leq W \\ W & \text{otherwise.} \end{cases} \tag{18}$$

Here, v_{tj} represents the velocity of the t^{th} particle on the j^{th} bit; y_{tj} is the position of the t^{th} particle on the j^{th} bit; p_{tj} is the best position of the t^{th} particle on the j^{th} bit; p_{gj} is the best position of the population on the j^{th} bit so far; r, r_1 and $r_2 \in \{0, 1\}$; c_1 and c_2 are the factors of learning, where $c_1, c_2 \geq 0$; $t = 1, 2, \ldots, N_{PB}$, $i = 1, 2, \ldots, M$.

3.5. **Termination rule.** The maximum number of iteration is used as the termination rule. They are N_{IT} and N_{IB} for the top-search and the base-search, respectively.

3.6. **The procedures of TSPSO.**

Step 1: Specify the parameters: the population size N_{PT} and N_{PB}, the maximum numbers of iteration N_{IT} and N_{IB}.

Step 2: Generate an initial top-pop with N_{PT} particles $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_{PT}}\}$.

Step 3: For particle $\mathbf{x}_k$ $(k = 1, \ldots, N_{PT})$.

Substep 3.1: Generate an initial base-pop with N_{PB} particles $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_{PB}}\}$.

Substep 3.2: For the anticipated base model, use the MCS process to calculate the fitness of each particle in base-pop, F^t_{AB} $(t = 1, \ldots, N_{PB})$.

Substep 3.3: Check the termination rule of the base-search. If it is satisfied, go to Step 4; otherwise go to Substep 3.4.

Substep 3.4: Update particles $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_{PB}}\}$, go to Substep 3.2.

Step 4: Use function (12) to calculate the fitness of the top model for each particle, F^k_T $(k = 1, \ldots, N_{PT})$.

Step 5: Check the termination rule of the top-search. If satisfied, go to Step 7; otherwise go to Step 6.

Step 6: Update particles $\mathbf{y}_t$ $(t = 1, \ldots, N_{PB})$, go to Step 3.

Step 7: Record the anticipated optimal solution $\hat{\mathbf{x}^*} = \{\hat{x^*_0}, \ldots, \hat{x^*_M}\}$ and $\{\hat{\mathbf{y}^*_1}, \ldots, \hat{\mathbf{y}^*_M}\}$. Here $\hat{\mathbf{y}}^*_i$ denotes the anticipated optimal base level solution under the anticipated top level optimal solution $\hat{x}^*_i$.

Step 8: For particle $\hat{\mathbf{x}^*}$.

Substep 8.1: Generate an initial base-pop with N_{PB} particles $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_{PB}}\}$.

Substep 8.2: For the base model, use function (14) to calculate the fitness of each particle, F^t_B $(t = 1, \ldots, N_{PB})$.

Substep 8.3: Check the termination rule of the base-search. If it is satisfied, go to Step 9; otherwise go to Substep 8.4.

Substep 8.4: Update particles $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_{PB}}\}$, go to Substep 8.2.

Step 9: Use function (12) to calculate the fitness of the top model, F_T.

Step 10: Output the optimal solution $\{x^*_0, \ldots, x^*_M\}$ and $\{\mathbf{y}^*_1, \ldots, \mathbf{y}^*_M\}$.

4. **Numerical Experiments.** In this section, two numerical examples are provided to illustrate the practical implementation of the model and the effectiveness and the reliability of TSPSO. The TSPSO approach of this study has been compared with two other approaches, GA and PSO.

4.1. **Example *I*.**

4.1.1. *The example.* In this example, the case when there are one owner and one partner is considered, so the number of member in the VE is 2, $(M + 1 = 2)$. The owner has to allocate the budget to the partner and itself, and the upper band of total budget is \$400 $(B_{\max} = 400)$. The risk loss of the owner is up to its budget and is given by a convex decreasing function

$$L_0(x_0) = 2000 \exp(-0.05x_0). \tag{19}$$

For owner, some information of partner is uncertain, such as the probability of risk occurrence p_{ij}. In this perspective, it can be taken as a random variable. In order to improve its information condition, the owner has to anticipate the probability of risk occurrence. However, it is difficult to know the real value, the owner can only know the distribution function of the anticipated probability of risk occurrence. It is assumed that $\hat{p_{ij}}$ follows a normal distribution function $N(p_{ij}, \sigma^2)$. p_{ij} is the expected value. In this example, p_{ij} is also the real probability of risk occurrence for the partner. The probability of risk occurrence for each risk factor is $p_{ij} = \{0.97, 0.90, 0.77, 0.70, 0.57, 0.47, 0.30, 0.22, 0.20, 0.07\}$, $i = 1$, $j = 1, 2, \ldots, N$. σ is the variance, $\sigma = 0.1$. The target benefit for partner is \$1,300,

$TB_i = 1300$, $i = 1$, which means that the risk management benefit of the partner must beyond $\$1,300$.

For the partner, there are 10 ($N = 10$) risk factors which have potential to threaten its safety, and 4 ($W = 4$) risk control strategies are provided for each risk factor. In this example, only one of the strategies will be selected for each risk factor or do nothing with it. The strategies are sequenced from low to high according to their effect. Generally, a stronger effect corresponds to a bigger index of strategy. The risk loss function is a convex decreasing function,

$$l_{ij}(y_{ij}) = 200\exp(-\chi_{ij}y_{ij}) \tag{20}$$

The parameters χ_{ij} ($i = 1, j = 1, 2, \ldots, N$) are used to describe the effect of various risk factors on risk loss, the value of the parameters are $\chi_{ij} = \{0.94, 0.87, 0.83, 0.73, 0.63, 0.50, 0.37, 0.33, 0.23, 0.1\}$, $i = 1, j = 1, 2, \ldots, N$. According to the specific form of risk loss function and the probability of risk occurrence, the initial risk loss for owner and the partner are both $\$2,000$, $L_i^{initial} = 2,000$, $i = 0, 1$. So the total initial risk loss of VE is $\$4,000$.

The partner has to pay for the cost of risk control strategies under its budgets. The cost of the strategy is assumed to be a concave increasing function of the corresponding strategy, which is approximated by:

$$C_{ij}(y_{ij}) = 50\left[1 - \exp\left(-\epsilon_{ij}y_{ij}\right)\right], \quad i = 1. \tag{21}$$

Here the parameters ϵ_{ij} are set according to the risk factors, $\epsilon_{ij} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, $j = 1, 2, \ldots, N$. Generally, a stronger effect strategy corresponds to a higher cost and a lower risk loss. From the form of the risk loss functions and the cost functions, it can be seen that an additional cost of selecting a stronger effect action yields a small decrease in the risk loss.

4.1.2. *Experimental results.* Firstly, the owner makes decision under its anticipation, the experimental results for Example I is listed in Table 1. After the owner, the partner makes the real decision, the experimental results for Example I is listed in Table 2. The maximize fitness value is used as the best experimental result in Tables 1 and 2. The owner anticipates the decision of the partner by its known information. The owner allocates the total budget \$394.15 to partner (\$256.52) and itself (\$137.63). It is anticipated that the partner will select the risk control strategies $\{4222100000\}$, and generates a benefit of risk management \$1328.20 which is beyond the target benefit $\$1,300$. The anticipated benefit of the whole VE is \$3152.43. After the anticipation, the owner sends the budget \$256.52 to the partner. The partner selects the optimal risk control strategies $\{3322100000\}$, and the benefit of partner is \$1318.37, which is lower than the anticipated result (\$1328.20). The real benefit of the whole VE is \$3142.60, which is also lower than the anticipated result (\$3152.43). The experimental results show that, the owner didn't know the real situation of partner, so the real decision result is not good enough like what the owner anticipated.

In the example, the coefficient η and λ in fitness function (12) are set $\eta = 10$ and $\lambda = 10$ respectively. The coefficient γ in fitness function (13) and (14) is set $\gamma = 100$ respectively. The learning factors c_1 and c_2 in Equations (15) and (17) are set $c_1 = 2.05$, $c_2 = 2.05$ and $c_1 = 2$, $c_2 = 2$ respectively [12, 13, 14]. The parameters setting of TSPSO is presented in Table 3. They are top-pop size N_{PT}, base-pop size N_{PB}, number of iterations for top-search N_{IT} and number of iterations for base-search N_{IB}.

The influence of the parameters on the fitness of best solutions are shown in Figures 2 and 3 respectively. They are top-pop size N_{PT}, base-pop size N_{PB}, number of iterations for base-search N_{IB}. While the influence of one parameter is shown, other parameters

Table 1. Anticipated decision results of Example I

Anticipated Decision Results	Value
Benefit of VE ($)	3152.43
Benefit of partner ($)	1328.20
Budget for members ($)	137.63 256.52
Total budget ($)	394.15
cost of partner ($)	248.47
strategy selected by partner	4 2 2 2 1 0 0 0 0 0

Table 2. Real decision results of Example I

Real Decision Results	Value
Benefit of VE ($)	3142.60
Benefit of partner ($)	1318.37
Budget for members ($)	137.63 256.52
Total budget ($)	394.15
Cost of partner ($)	249.57
Strategies selected by partner	3 3 2 2 1 0 0 0 0 0

Table 3. Parameters setting of TSPSO for Example I

Parameters	N_{PT}	N_{PB}	N_{IT}	N_{IB}
Values	200	80	80	40

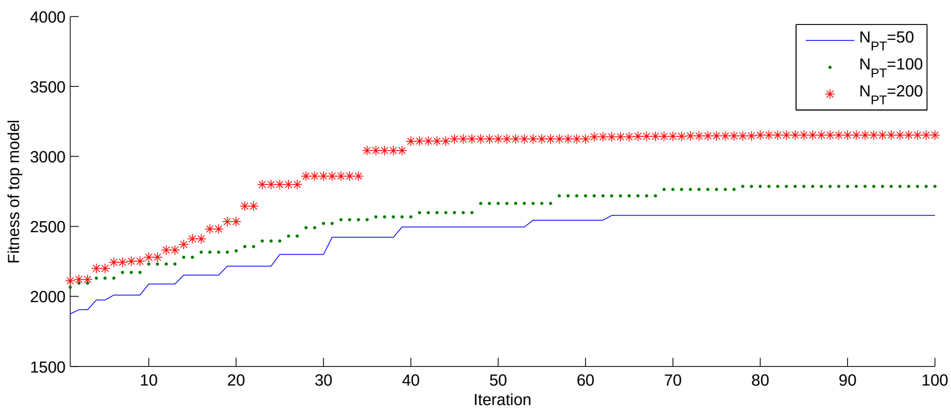

Figure 2. The influence of N_{PT} on the fitness

are fixed as in Table 3. Figures 2 and 3 illustrate the convergence characteristics of the proposed TSPSO with different population size respectively. When the top-pop size increases, the TSPSO can find better solutions and provide an indication for its robustness. By comparing Figures 2 and 3, a finding is that the top-pop size makes a stronger influence on the fitness than base-pop size. In other words, the top-search is more important for the convergence of TSPSO with respect to the cooperative function of top-search. Figure 4 presents the convergence process of the three heuristic methods. The average fitness value of 10 runs of each algorithm is used. Among the three heuristic methods, TSPSO can search out the optimal solution and generate the optimal efficient frontier more frequently than other two optimizers. In early stage of the running, PSO converged most rapidly,

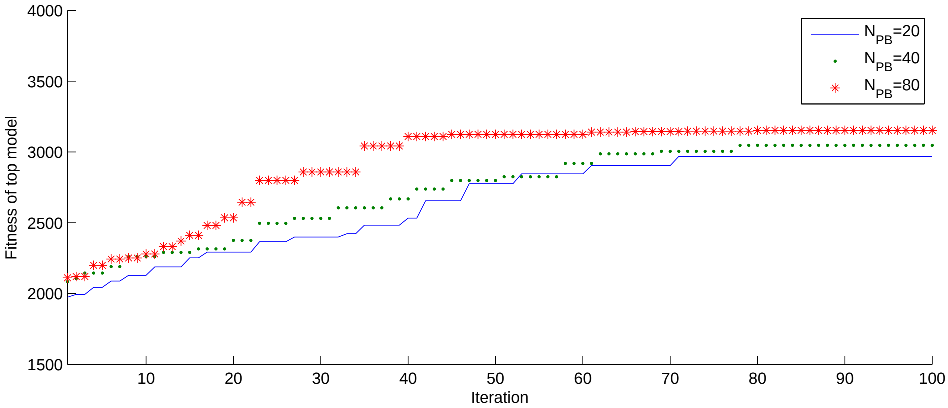

FIGURE 3. The influence of N_{PB} on the fitness

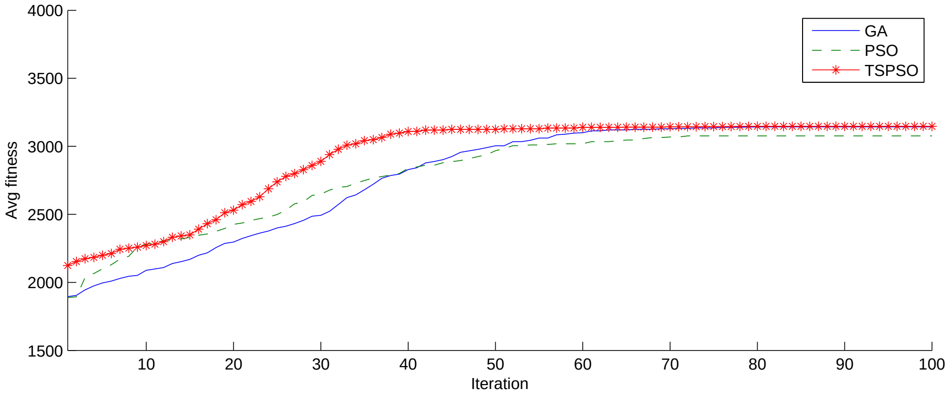

FIGURE 4. The convergence process of the three heuristic methods

TABLE 4. The Min, Max, Avg and Std of the three heuristic methods for Example I

	GA	PSO	TSPSO
Min	3129.21	2934.63	3141.84
Max	3152.43	3139.12	3152.43
Avg	3142.83	3076.51	3147.3
Std	9.0131	18.0121	4.0069

but may encounter premature convergence. GA convergence slowly than the other two optimizers in the early stage but GA find its best solution which is mostly as good as TSPSO, at the end of the search.

Since GA, PSO and TSPSO are stochastic optimization methods, a statistical evaluation of their performance is necessary. Here the experiments are repeated 10 times with different random seeds. The minimum (Min), maximize (Max), average (Avg) and standard deviation (Std) of fitness value from the best solutions obtained are reported in Table 4. GA and TLPSO find the best solution, PSO makes the worst performance on the maximum fitness. TSPSO has the best searching ability and reliability with respect to the Avg and Std.

4.2. **Larger numerical example: Example II.** To demonstrate the effectiveness of the proposed algorithm, a larger size problems are provided.

4.2.1. *The example.* For example II, there are also one owner and one partner. The deference comes from the partner, more risk factors are considered. It is supposed that there are 30 ($N = 30$) risk factors which have potential to threaten its safety, while risk control strategies for each risk factor are still 4 ($W = 4$). The owner has to allocate the budget to the partner and itself, and the upper band of total budget is \$800. The risk loss of the owner is up to its budget and is given by a convex decreasing function, see the risk loss function of the owner in Example I. The target benefit for the partner is $\$3,900$, $TB_i = 3900$, $i = 1$. The risk loss functions for the partner is a convex decreasing functions, which is the same as in Example I. The initial risk loss for owner is $\$2,000$, $L_i^{initial} = 2,000$, $i = 0$. The initial risk loss for partner is $\$6,000$, $L_i^{initial} = 6,000$, $i = 1$. So the total initial risk loss of VE is $\$8,000$ for Example II. The partner has to pay for the cost of risk control strategies under its budgets. The cost function of the corresponding strategies, see the cost function in Example I.

4.2.2. *Experimental results.* The parameters setting of TSPSO for Example II are presented in Table 5. The top level optimization problem is a continuous one, while the base level optimization problem is a discrete one. So the sizes of the base level optimization problem without respect to the constraints for Example I and II are $1.05 + E06$ and $1.15 + E18$ respectively. For Example II, the experiment is also repeated 10 times with different random seeds, the Min, Max, Avg and Std of fitness value from the best solutions of the three heuristic methods are recorded in Table 6. Comparing Table 6 with Table 4, GA find a better solution than PSO's, and make a better reliability than PSO's for the two examples. while the size of the problem increase, it can be seen that the TSPSO keeps its best performance, the best solution, and the Std of the fitness, especially for the larger size examples.

Table 5. Parameters setting of TSPSO for Example II

Parameters	N_{PT}	N_{PB}	N_{IT}	N_{IB}
II	240	100	80	40

Table 6. The Min, Max, Avg and Std of the three heuristic methods for Example II

	GA	PSO	TSPSO
Min	5621.39	5546.14	5766.33
Max	5798.22	5739.69	5926.12
Avg	5747.64	5656.87	5862.03
Std	17.3002	28.0043	11.7403

On the other hand, the computational time of each algorithm on the two examples are compared. Figure 5 shows the computational time of the three heuristic methods on Example I and II respectively. GA and PSO are shown to be the least time consuming method, Although the TSPSO algorithm always seems to be more time consuming than the others. However, in Table 7, TSPSO can find out the better solution with less times of increment on computational time than the other two optimizers due to its fast convergence speed. From Example I to II, the times of increment on computational time of GA, PSO

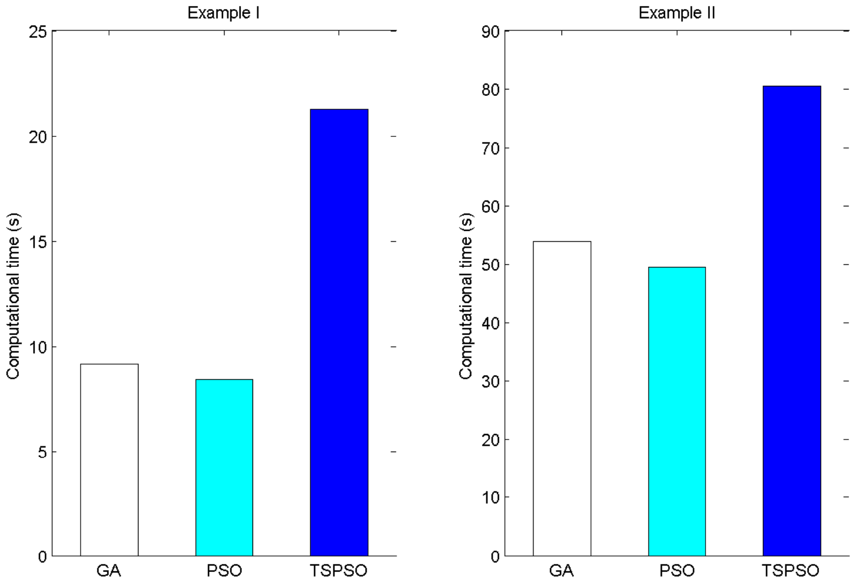

FIGURE 5. The computational time of the three heuristic methods for Example I and II

TABLE 7. The times of increment on computational time from Example I to II

	GA	PSO	TSPSO
The times of increment	5.90	5.89	3.79

and TLPSO are 5.90, 5.89 and 3.79 respectively, which is under the same increment of the problem sizes.

5. **Conclusions.** In this paper, a TSPSO is proposed for risk management of VE. The risk management is formulated as a two-level optimization problem with hierarchical structure. Taking the characteristics of the optimization problem into account, TSPSO is proposed, which has two level searching process and multiple swarms, and which is an improvement on the previous single searching process and single swarm optimizers. Numerical example is given, GA and PSO are also designed for comparing with TSPSO. The simulation results show that, TSPSO can search out the best solutions rapidly, but GA and PSO cannot. Therefore, TSPSO is a very efficient and reliable method to solve this kind of optimizing problem with hierarchical structure and high time complexity.

Acknowledgment. This work is partially supported by the Foundation of Northeastern University at Qinhuangdao under Grant No. XNB201404. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] R. L. Kliem and I. S. Ludin, *Reducing Project Risk*, England, Gower, 1997.
[2] D. W. Wang, K. L. Yung and W. H. Ip. A heuristic genetic algorithm for subcontractor selection in a global manufacturing environment, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.13, no.2, pp.189-198, 2001.

[3] W. Ip, M. Huang, K. Yung and D. Wang, Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise, *Computers and Operations Research*, vol.30, pp.213-231, 2003.
[4] M. Huang, W. H. Ip, H. Yang, X. Wang and H. C. W. Lau, A fuzzy synthetic evaluation embedded tabu search for risk programming of virtual enterprises, *International Journal of Production Economics*, vol.116, no.1, pp.104-114, 2008.
[5] F. Tao, L. Zhang, Z. H. Zhang and A. Y. C. Nee, A quantum multi-agent evolutionary algorithm for selection of partners in a virtual enterprise, *CIRP Annals – Manufacturing Technology*, vol.59, pp.485-488, 2010.
[6] M. Huang, F. Q. Lu, W. K. Ching and T. K. Siu, A distributed decision making model for risk management of virtual enterprise, *Expert Systems with Applications*, vol.38, no.10, pp.13208-13215, 2011.
[7] Y. C. Shao, X. J. Yao, L. W. Tian and H. N. Chen, A multiswarm optimizer for distributed decision making in virtual enterprise risk management, *Discrete Dynamics in Nature and Society*, 2012.
[8] M. Huang, X. W. Wang, F. Q. Lu and H. L. Bi, A coordination study in risk management of supply chain organized as virtual enterprise, *Mathematical Problems in Engineering*, 2013.
[9] A. Fernandez and S. Gomez, Portfolio selection using neural networks, *Computers & Operations Research*, vol.34, pp.1177-1191, 2007.
[10] J. W. Gao and Z. H. Chu, A new particle swarm optimization based on MATLAB for portfolio selection problem, *International Journal of Modeling, Identification and Control*, vol.9, pp.206-211, 2010.
[11] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
[12] C. C. Tseng, J. G. Hsieh and J. H. Jeng, Active contour model via multi-population particle swarm optimization, *Expert Systems with Applications*, vol.36, pp.5348-5352, 2009.
[13] F. Valdez, P. Melin and O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, *Appl. Soft Comput.*, vol.11, no.2, pp.2625-2632, 2011.
[14] A. Buthainah, *Multi-Phase Particle Swarm Optimization*, Ph.D. Thesis, Syracuse University, Italia, 2002.
[15] W. L. Du and B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, *Information Sciences*, vol.178, pp.3096-3109, 2008.
[16] S. M. Seyedpoor, Structural damage detection using a multi-stage particle swarm optimization, *Expert Systems with Applications*, vol.14, no.3, pp.533-549, 2011.