J Intell Manuf (2015) 26:899–910
DOI 10.1007/s10845-014-0905-8

# GA–SA/CPM/Markov based dynamic risk-management planning for virtual enterprises

**Guanjie Jiang · Min Huang · Chunhui Xu · Xingwei Wang**

Received: 7 November 2013 / Accepted: 19 March 2014 / Published online: 18 April 2014


**Abstract** A virtual enterprise (VE) is always in an environment with unpredictable change and dynamic markets. Therefore, a VE is more susceptible to risks. Minimizing risk in the operation of a project and ensuring success are major issues to concern in VEs. This paper develops a novel three-level dynamic risk-management planning model for VEs focusing on project organization mode and dynamic features of risk with the objective to maximize the completion probability under the constraints of cost, due date, and quality. The first level adopts non-linear integer programming techniques, the second level is about risk evaluation for the whole project based on network analysis, and the third level is on a Markov process based single process risk evaluation. An algorithm of integrated genetic algorithm plus simulated annealing/critical path method/Markov (GA–SA/CPM/Markov) is then designed to solve the problem and compared with GA and SA respectively. Experimental results show that the proposed algorithm is effective and the three-level model can deal with dynamic risks for VEs.

**Keywords** Virtual enterprise · Dynamic risk-management planning · Genetic algorithm · Simulated annealing · Critical path method · Markov process

G. Jiang · M. Huang (✉) · X. Wang
College of Information Science and Engineering, Northeastern University, Shenyang 110819, Liaoning, China
e-mail: mhuang@mail.neu.edu.cn

G. Jiang · M. Huang · X. Wang
State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, Liaoning, China

G. Jiang
School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China

M. Huang
Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang 110819, Liaoning, China

C. Xu
Department of Risk Science in Finance and Management, Chiba Institute of Technology, Chiba 275-0016, Japan

## Introduction

Today's enterprises face a dynamic global market in which market opportunities can arise and disappear again within a short time duration. Traditional enterprises' ability remains limited in adapting themselves to these changes (Katzy and Dissel 2001). Thus, virtual enterprises (VEs) have emerged and become a major entrepreneurial mode with specialization and flexibility (Park and Favrel 1999; Bernus and Nemes 1999; Martinez et al. 2001; Cardoso and Oliveira 2005; Camarinha-Matos and Pantoja-Lima 2001). VE can be defined as a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities and towards mutual goals through strong cooperation supported by information and communication technology (ICT) (Camarinha-Matos and Afsarmanesh 2003; Kierzkowski 2005; Soares et al. 2000).

On one hand, VE is becoming essential approach to meet the market's requirements for quality, responsiveness, and customer satisfaction (Chen et al. 2010). On the other hand, enterprises in a VE face more risks connected with disruption, inventories and schedule, quality, technology, price, and information leakage (Treleven and Schweikhart 1988; Zhang et al. 2012). Therefore, how to reduce the effect of risks on VE and manage risks is a key problem to approach in VEs so as to ensure success.

Total risk management process consists of risk assessment and risk management (Haimes 2009). Risk assessment involves risk identification, risk analysis, and risk prioritization. Based on risk assessment, risk management involves risk-management planning, risk resolution, and risk monitoring (Boehm 1991). As an important stage in risk management, risk-management planning helps to mitigate risks and reduce their detrimental effects to the least through selecting the most appropriate strategies. In this paper, the main research problems are:

- How to describe the problem of dynamic risk-management planning in a VE quantificationally?
- What is the relationship between the risk of VE and that of the process in a VE, and how to express it?
- How can the effects of the dynamic risks on the processes be described?

In order to deal with these problems, a three-level model for dynamic risk-management planning is established in this paper. The first-level model describes the problem of dynamic risk-management planning in a VE with the objective to maximize the minimum completion probability of all processes under the constraints of project investment, due date, and quality requirement. The second-level model is to make holistic risk evaluation of the project through network analysis, which describes the relationship between the risk of VE and that of the process in VE. The third-level model makes use of Markov process to evaluate the risk of a single process where the effect of the dynamic risk is described. To deal with the three-level model, an algorithm of integrated genetic algorithm plus simulated annealing/critical path method/Markov (GA–SA/CPM/Markov) is designed. Simulation shows the effectiveness of the proposed model and algorithm.

The main contributions of this paper are twofold. First, the description of the dynamic risk-management planning problem is introduced. The proposed model provides a formal description for dynamic risk-management planning problem of VE. This has the potential to be an efficient quantitative tool for risk management in the virtual global business environment. Second, the algorithm of GA–SA/CPM/Markov is proposed to solve this problem. Even though this study considers the subject from VEs' viewpoint, the method can be extended to other projects with dynamic risk, such as construction project.

The rest of this paper is organized as follows. The relevant literature is reviewed in section "Literature review". Section "Problem description and the three-level model of dynamic risk-management planning" describes the problem and proposes a three-level model for dynamic risk-management planning with a focus on a project's organization mode and a VE's dynamic characteristics. In section "An integrated GA–SA/CPM/Markov algorithm", a GA–SA/CPM/Markov algorithm is developed for the model. Examples and experimental results are included in section "Numerical analysis". The conclusions are given in section "Conclusion".

## Literature review

While they bring many benefits, VEs also imply some unavoidable risks. These risks come from external and/or internal environments. The former includes market, financial, political, and natural risks. The latter includes competency, relational, investment, and operational risks (Li and Liao 2007). Hence, risk management of a VE become imperative for researchers and managers of enterprises.

As a VE is a complex system temporarily composed of many enterprises responding to market opportunities, the traditional risk model no longer works (Park and Favrel 1999; Hallikas et al. 2002; Kliem and Ludin 1997; Bier et al. 1999; Haimes 2009).

Some researchers focus on risk identification and evaluation. Hallikas et al. (2002) described a conceptual framework for analyzing and assessing a production network's risk from both a buying company and a supplying company's viewpoints. Specifically, they present two approaches. One is internal audit, the other is computer aided cause and effect analysis. Yet this research is restricted to qualitative information. Li and Liao (2007) identified various kinds of risk factors affecting the operation of alliance and set up three measurements (risk occurrence likelihood, consequence severity, and risk control degree) expressed by trapezoidal fuzzy numbers for these risks. They then proposed a new risk evaluation approach based on the framework of the evidential reasoning. Huang et al. (2008) focused on VEs' two main features, i.e., project mode and uncertain factors. They established the fuzzy synthetic evaluation embedded nonlinear integer programming model of risk-management planning for the VE and presented a tabu search algorithm with an embedded fuzzy synthetic evaluation for the model.

Risk problem in partner selection of VEs has received research attentions. Ip et al. (2003) studied the optimization model for minimizing risk in partner selection while ensuring the due date of a project for a VE. 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 was developed to solve the problem. Chan and Kumar (2007) gave a fuzzy extended analytic hierarchy process based methodology to tackle different decision criteria like cost, quality, service performances, and supplier's profile including the risk factors involved in the selection of global supplier in the current business scenario. Mun et al. (2009) proposed a trust evaluation method with fuzzy inference system in order to support enterprise collaboration and maximize its satisfaction.

Other researchers focus on risk management on the basis of VEs' distributed characteristics. Huang et al. (2011) developed a distributed decision-making (DDM) model for risk management. The model has two levels, i.e., the top and base ones, which describe the decision processes of the owner and partners, respectively. A particle swarm optimization (PSO) algorithm is then designed to solve the resulting optimization problem. In order to solve the proposed DDM model effectively, Chen et al. (2010) applied two powerful artificial intelligence optimization techniques known as evolutionary algorithms and swarm intelligence.

All above mentioned literature concerns about static risk either under a centralized or DDM circumstance but neglects risks' dynamic characteristics. In reality, during completing a project, risks are involved in many processes. Due to the uncertainty of external environment and enterprise itself such as the change of weather, demand fluctuation, and failures in production equipment, risks are usually dynamic. To reduce dynamic risks' effect on a process, how to select completion time for each process is the main decision problem.

## Problem description and the three-level model of dynamic risk-management planning

Assume a VE carries out a project consisting of multiple processes. Each process is divided into several stages and each stage consists of various states affected by risk factors. For any state, the current one moves to the next one with a certain probability. The process finally attains either a success or a failure state. Specifically, the transition probability depends solely on the present state. This kind of "memorylessness" is called Markov property. Moreover, each process of a project has different completion time choices. As a consequence, it suffers from different influences of risk factors. For any state, its relevant transition probabilities are different. Their values depend on the selected completion time which causes different completion probability, cost, and quality. Decision makers can choose an appropriate completion time according to their actual circumstances so as to attain the objective risk performance in terms of the whole project's completion time, probability, cost, and quality. For a project, its constraints include the investment, due date, and quality requirement. This paper aims to determine all processes' completion time as well as maximize the minimum completion probability after a project commences.

### Notations

| | |
|---|---|
| $t_i$ | Completion time of process $i$ |
| $B_w(T)$ | Completion probability of the project, which is the minimum among the completion probabilities of all processes |
| $C_w(T)$ | Completion cost of the project |
| $T_w(T)$ | Completion time of the project |
| $M_i(t_i)$ | Completion quality coefficient corresponding to $t_i$ |
| $C_i(t_i)$ | Completion cost of process $i$ corresponding to $t_i$ |
| $M_{0i}$ | Quality coefficient required by process $i$ $M_{0i} = \mu_i - \sigma_i$, $\mu_i$ is the mean value of process $i$'s quality coefficients and $\sigma_i$ is the corresponding standard deviation |
| $N$ | Number of processes in the project |
| $T_0$ | Project due date |
| $C_0$ | Required completion cost of the project, i.e., the project investment |
| $b_i(t_i)$ | Completion probability of process $i$ corresponding to $t_i$ |
| $K$ | Set of processes on a certain critical path |

The dynamic risk-management planning process is described by a three-level model which is elaborated as follows:

The first-level model M1:

$$\max B_W(T) \tag{1}$$

$$\text{s.t} \quad C_W(T) \leq C_0 \tag{2}$$

$$T_W(T) \leq T_0 \tag{3}$$

$$M_i(t_i) \geq M_{0i} \tag{4}$$

$$T = \{t_1, t_2, \ldots, t_N\}, t_i \in \left[t_i^1, t_i^2, \ldots, t_i^{I_i}\right], i = 1, 2, \ldots N \tag{5}$$

where $B_W(T)$, $C_W(T)$, and $T_W(T)$ are determined by the second-level model and $M_{0i}$ is determined by the third model, respectively. The first level is a non-linear integer programming. It helps to determine the completion time of the project under the constraints of project investment, due date, and quality requirement. Its major focus is to maximize the minimum completion probability of all processes in a project.

The second-level model M2:

Under the assumption that each process' precedence is known and can be elaborated by a network, the whole project's completion probability, cost, time, and quality are

$$B_W(T) = \min\{b_i(t_i), i = 1, 2, \ldots, N\} \tag{6}$$

$$C_W(T) = \sum_{i=1}^{N} C_i(t_i) \tag{7}$$

$$T_W(T) = \sum_{i \in K} t_i \tag{8}$$

$$M_W(T) = \min\{M_i(t_i), i = 1, 2, \ldots N\} \tag{9}$$

where $b_i(t_i)$, $C_i(t_i)$, and $M_i(t_i)$ are determined by the third level.

We use the second level to evaluate the risk of the project through network analysis. In this level, a project's completion time is determined by CPM. A project's cost is the sum of all processes' cost. A project's completion probability and quality is the minimum completion probability and the minimum quality coefficient among all processes, respectively.

The third-level model M3:

Cowing et al. (2004) presented and illustrated a dynamic probabilistic model to describe the long-term evolution of such a system through the different phases of operation, shutdown, and all accidents if any, where a Markov model is used to track the evolution of the system and its components through different performance phases. Hereby, the third level evaluates each process' risks with a Markov process which describes the state transition. It determines the completion probability, cost, and quality of processes when completion times are distinct.

In the third-level model, suppose that process $i$ operates in two stages: preparation and operation. According to the influences from risk factors, process $i$ is divided into $n_i$ states constituting the state space $S_i$. $S_i = \left\{x_1^i, x_2^i, \ldots, x_{n_i}^i\right\}$ is a Markov process. Suppose there is one state $x_1^i$ in the preparatory stage, and $n_i - 3$ states $\left\{x_2^i, x_3^i, \ldots, x_{n_i-2}^i\right\}$ exist in operation stage. In the end, the process reaches a success state $x_{n_i-1}^i$ or a failure state $x_{n_i}^i$ and they are called absorbing states. Suppose there are $n_i - 2$ irreducible states and the process operation can be described by applying the properties of Markov Chain. State transition is shown in Fig. 1.

Since a process' operation is dynamic, a transition's action has to be detailed with a probability. This paper assumes that such probability is a function of time which depends on the process' completion time. A process' completion time varies owing to the decision maker's choice. When the completion time of process $i$ is $t_i$, the state transition matrix is as follows:

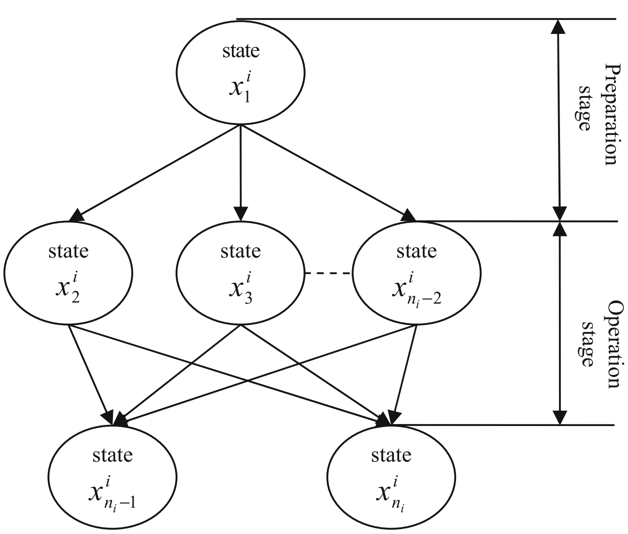


**Fig. 1** Transition of states

$$P_i\left(t_i\right) = \begin{bmatrix} Q_i\left(t_i\right) & R_i\left(t_i\right) \\ \varphi & I \end{bmatrix}$$

$$= \begin{array}{cc} & \begin{array}{ccccccc} x_1^i & x_2^i & x_3^i & \cdots & x_{n_i-2}^i & x_{n_i-1}^i & x_{n_i}^i \end{array} \\ \begin{array}{c} x_1^i \\ x_2^i \\ x_3^i \\ \vdots \\ x_{n_i-2}^i \\ x_{n_i-1}^i \\ x_{n_i}^i \end{array} & \begin{bmatrix} 0 & p_{12(t_i)}^i & p_{13(t_i)}^i & \cdots & p_{1,n_i-2}^i(t_i) & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & r_{2,n_i-1}^i(t_i) & r_{2,n_i}^i\left(t_i\right) \\ 0 & 0 & 0 & \cdots & 0 & r_{3,n_i-1}^i(t_i) & r_{3,n_i}^i\left(t_i\right) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{n_i-2,n_i-1}^i(t_i) & r_{n_{i-2,n_i}}^i\left(t_i\right) \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{10}$$

where $Q_i\left(t_i\right)$ represents the relationship among the irreducible states of process $i$; $R_i\left(t_i\right)$ indicates the relationship between the irreducible and absorbing states; $I$ is a $2 \times 2$ identity matrix; $\varphi$ is a $2 \times (n_{\mathrm{i}} - 2)$ zero matrix.

A parameter $p_{hl}^i\left(t_i\right)$ represents the probability from state $x_h^i$ entering into an irreducible state $x_l^i$; $r_{hl}^i\left(t_i\right)$ represents the probability that state $x_h^i$ enters into an absorbing state $x_l^i$; then, we have $\sum_{l=1}^{n_i-2} p_{hl}^i\left(t_i\right) + \sum_{l=n_i-1}^{n_i} r_{hl}^i\left(t_i\right) = 1, 0 \leq p_{hl}^i\left(t_i\right) \leq 1, 0 \leq r_{hl}^i\left(t_i\right) \leq 1$.

From the above analysis, our problem's objective is nonlinear and with discrete variables. Furthermore, it is NP-hard and with nonconvex space. As well, our objective function is neither continuous nor differentiable. Such models cannot be solved by general mathematical techniques. As a resolution, this paper proposes a meta-heuristic algorithm to solve them.

## An integrated GA–SA/CPM/Markov algorithm

By taking advantage of the concerned problem's characteristics, we propose an integrated GA–SA/CPM/Markov algorithm such that the problem can be handled at different levels. A GA–SA algorithm is designed to solve the optimization problem (Kirkpatrick et al. 1983; Michalewicz 1994). CPM is used on the second level (Carter and Price 2001). By using Markov property, the calculation is done on the third level (Ibe 2008; Cowing et al. 2004).

### To solve the first-level model based on GA–SA

For the first-level model with formulas (1)–(5), the solution space is $\prod_{i=1}^{N} I_i$ (without considering the constraints). Obviously, the solution space is quite large even for a small-size problem. For example, for a project with 15 processes and each process with 5 candidate times, the solution space is $3.052 \times 10^{10}$. Therefore, a meta-heuristic algorithm is needed to deal with this problem.

GA is advantageous thanks to its robust-optimization feature and global-search capability for non-linear problems like combination optimization (Goren et al. 2010). GA has strong exploration ability while it is weak in exploitation ability.

However, SA is a highly effective stochastic method based on local search to solve combinatorial optimization problem (Yu et al. 2010; Yen et al. 2004). This paper proposes a GA–SA hybrid framework where GA explores better solutions by means of its strong global search ability and SA refines these solutions further in order to find the best solution.

*Design of GA–SA for dynamic risk-management planning*

**Parameters**

| | |
|---|---|
| $c_0$ | Initial temperature |
| $c_f$ | Stopping temperature |
| $r$ | Cooling rate |
| $M$ | Number of internal cycles |
| $GN$ | Maximum number of generations |
| $NP$ | Population size |
| $P_0$ | Initial population |
| $P_i$ | Population of the $i$th generation |
| $p_c$ | Crossover probability |
| $p_m$ | Mutation probability |

(1) Code design
One chromosome represents only one solution in the search space. In this paper, it denotes a complete decision for the proposed problem. Here, let the integer number string $u = [u_1, u_2, \ldots, u_N]$ represent a chromosome where $u_i, i \in \{1, 2, \ldots N\}$, is the integer from 1 to $I_i$, showing that process $i$ chooses completion time $t_i^{u_i}$. Therefore, $u = [u_1, u_2, \ldots, u_N]$ refers to a decision. For example, (2,1,3,1,2,1,3,1,1,2,4,1,1) represents the sequential corresponding number of time choice of a 13-process project. The number 2 in the first place of the code set means the first process chooses the second completion time choice, and so on.

(2) Initialization

**Step 1**: Set the number of population as $NP$.
**Step 2**: Let $S = 0$ and $i = 0$, go to **Step 3**.
**Step 3:** Randomly generate an integer $b \in [1, I_\mathrm{i}]$ and let $u_i = b$, go to **Step 4**.
**Step 4**: $i = i + 1$. If $i < N$, go to **Step 3;** otherwise a complete chromosome $u^S = [u_1, u_2, \cdots, u_N]$ is obtained, go to **Step 5**.
**Step 5**: $S = S + 1$. If $S < NP$, go to **Step 3;** otherwise stop.

(3) Fitness function
Since the objective function of the model is to maximize the minimum completion probability of the process, the fitness function for the $j$th solution of GA is:

$$FIT(j) = B_\mathrm{W}(T) \tag{11}$$

According to the objective values of the chromosomes, we give a non-increasing order relationship among the $NP$ chromosomes to rearrange the chromosomes. Then, the fitness of each chromosome can be calculated by the evaluation function given as

$$eval(i) = a(1-a)^{i-1}, \quad i = 1, 2, \ldots, NP \tag{12}$$

where $a \in (0, 1)$ is a parameter given in advance.
The fitness function for the $j$th solution of SA is:

$$f(j) = \frac{1}{B_\mathrm{W}(T)} \tag{13}$$

(4) Crossover and mutation
Here we adopt one-cut-point crossover. First, two chromosomes are picked from the population at random. Second, two children can be available by randomly generating an integer in the range $[1, N]$ as the crossover point and exchanging the parents' information.
One-bit mutation is used in this paper. First, randomly choose a chromosome from the individuals by crossover. Second, generate an integer in the range $[1, N]$ as the mutation bit. Third, randomly select an integer in the range $[1, I_i]$different from the current time choice of process $i$.
For an infeasible solution, we replace it with a new individual generated randomly in order to maintain the quality and diversity of population.

(5) Local refinement
In order to improve the exploitation ability, SA is used as a local search method embedded into the framework of GA, shown in Fig. 2.

Local search is performed on the offspring after crossover and mutation with the probability $\beta$ to save runtime and maintain the diversity of population. The neighborhood solution $h$ is selected by randomly choosing an integer in the range $[1, N]$ and then randomly generating an integer in the range $[1, I_i]$ different from the current choice. If the solution $h$ is infeasible, then generate a neighborhood solution again with the same method.

- Cooling function

In our problem the following method to reduce the temperature has been used:

$$c := r \cdot c \tag{14}$$

where $r$ is a cooling rate in the range [0,1]. $c$ is repeated from initial temperature $c_0$ to stopping temperature $c_f$.

**Fig. 2** Pseudo-code of SA algorithm

```
Begin
if (rand(0,1) < β) then //execute SA to the individual j
  set initial temperature c_0
  m = 1
  calculate fitness value f(j) according to formula (13), and set it as current
  best solution j* = j, u* = u(j), f* = f(j)
  while(c > c_f) //execute the following
    while(m < M)
      randomly generate a neighborhood solution h, switch to course P2;
      determine B_W(T), C_W(T), T_W(T), M_W(T) and M_i(t_i);
      calculate Δf = f(h) − f(j);
      if (Δf < 0) then
        u(j) = u(h)
      endif
      else if (Δf > 0) then
        u(j) = u(h) with probability exp(−Δf/c) > ξ
      endif
      if (f(j*) < f(h)) then
        renew the current best solution j* = h, u* = u(h), f* = f(h)
      endif
      m = m + 1
    endwhile
    c = r · c
  endwhile
endif
end
```

- The control of internal cycle

The given number $M$ is used for the control of internal cycle. Hereby, $M = 10$.

- Termination criterion

The given stopping temperature $c_f$ is set as the termination criterion.

*Procedures of the algorithm*

The whole algorithm is composed of two main procedures:

(1) A good population is generated by GA emphasizing particularly on global search.

(2) Optimal adjustment by SA emphasizes particularly on local search.

The whole procedures of the GA–SA are given in Fig. 3:
Course P1:

To solve the second-level model based on CPM

The key thing in model M2 is to fix the critical process followed by the determination of $B_W(T)$, $C_W(T)$, $T_W(T)$, and $M_W(T)$. The procedure is shown below.
Course P2:

**Step 1**: Calculate the earliest starting time $b'_i$ and the earliest completion time $c'_i$ for process $i$ from $i = 1$ to $N$;

**Fig. 3** Pseudo-code of GA–SA algorithm

**Begin**

initialize algorithm parameters $NP, GN, p_c, p_m, c_0, c_f, r, M$

generate initial population $P_0$,

$u(j)$=$[u_1(j), u_2(j), \cdots, u_N(j)], j = 1,2,\cdots, NP$

let $t_i = t_i^{u_i(j)}(i = 1,2,\cdots,N)$, switch to course P2 and determine $B_W(T), C_W(T), T_W(T)$, and $M_W(T)$

evaluate $P_0$ and find the best value $FIT^*$and the best solution $u^*$

$k = 0$

**repeat**

crossover and mutation

execute the local search method

generate $P_{k+1}$ according to elite preserving strategy and roulette wheel method

$k = k + 1;$

**until** a stop condition is met

**End**

$$b'_i = \begin{cases} 0, & P_i^B = \emptyset \\ \max\left\{c'_j, \forall j \in P_i^B\right\}, & P_i^B \neq \emptyset \end{cases} \tag{15}$$

$$c'_i = b'_i + t_i \tag{16}$$

where $P_i^B$ is the immediate predecessor process set before process $i$.

**Step 2**: Determine the latest completion time $c_i$ and the latest starting time $b_i$ for process $i$ from $i = N$ to 1.

$$c_i = \begin{cases} c'_N, & P_i^A = \emptyset \\ \min\left\{b_j, \forall j \in P_i^A\right\}, & P_i^A \neq \emptyset \end{cases} \tag{17}$$

$$b_i = c_i - t_i \tag{18}$$

where $P_i^A$ is the immediate successor after process $i$.

**Step 3**: For process $i$ from $i = 1$ to $N$, process $i$ on the critical path $\pi_h$if $c_i = c'_i$; or else, not on the critical path $\pi_h$; determine any critical path $\pi_k$.

**Step 4**: Take $t_i$ as the parameter and switch to course P3.

**Step 5**: Calculate $B_W(T), C_W(T), T_W(T)$, and $M_W(T)$ by formulas (6)–(9); return project parameters $B_W(T), C_W(T), T_W(T), M_W(T)$, and process parameters $M_i(t_i)$ to course P1.

In summary, **Steps 1-3** determine the critical path by calculating the earliest and latest times of process $i$. **Step 4** determines process $i$'s completion probability, cost, and quality coefficients. **Step 5** determines the project's completion probability, time, cost, and quality coefficients.

To solve the third-level model based on Markov process

It determines process $i$'s completion probability, cost, and quality with treatment time $t_i$ by the application of Markov process.

Course P3:

**Step 1**: Calculate the base matrix $F_i(t_i)$ and absorbing matrix $B_i(t_i)$ by applying the properties of Markov Chain:

$$F_i(t_i) = (I - Q_i(t_i))^{-1}$$
$$= \begin{bmatrix} 1 & -p_{12}^i(t_i) & -p_{13}^i(t_i) & \cdots & -p_{1,n_i-3}^i(t_i) \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}^{-1} \tag{19}$$

$$B_i(t_i) = F_i(t_i) \cdot R_i(t_i)$$
$$= \begin{bmatrix} 1 & -p_{12}^i(t_i) & -p_{13}^i(t_i) & \cdots & -p_{1,n_i-3}^i(t_i) \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} 0 & 0 \\ r^i_{2,n_i-1}(t_i) & r^i_{2,n_i}(t_i) \\ r^i_{3,n_i-1}(t_i) & r^i_{3,n_i}(t_i) \\ \vdots & \vdots \\ r^i_{n_i-2,n_i-1}(t_i) & r^i_{n_i-2,n_i}(t_i) \end{bmatrix}$$

$$= \begin{bmatrix} b^i_{11}(t_i) & b^i_{12}(t_i) \\ b^i_{21}(t_i) & b^i_{22}(t_i) \\ b^i_{31}(t_i) & b^i_{32}(t_i) \\ \vdots & \vdots \\ b^i_{n_i-2,1}(t_i) & b^i_{n_i-2,2}(t_i) \end{bmatrix} \quad (20)$$

**Step 2**: From the definition of absorbing matrix, the first element $b^i_{11}(t_i)$ of the first row of $B_i(t_i)$ is process $i$'s completion probability. Then

$$b_i(t_i) = b^i_{11}(t_i) \quad (21)$$

**Step 3**: Calculate the average cost $C_i(t_i)$ of process $i$:
Let $f_i$ represent the occupied cost vector of process $i$'s whole states such that each member $f_i\left(x^i_h\right)$ represents the occupied cost in state $x^i_h$, then

$$C_i(t_i) = \left[f_i\left(x^i_1\right), f_i\left(x^i_2\right), f_i\left(x^i_3\right), \ldots, f_i\left(x^i_{n_i-2}\right)\right] \cdot \begin{bmatrix} 1 \\ p^i_{12}(t_i) \\ p^i_{13}(t_i) \\ \vdots \\ p^i_{1,n_i-3}(t_i) \end{bmatrix} \quad (22)$$

**Step 4**: Calculate the quality coefficient $M_i(t_i)$ of process $i$:
Let $m_i$ represent the corresponding quality vector of process $i$'s states, $m_i\left(x^i_h\right)$ is the quality coefficient at state $x^i_h$, and $m_i = \left[m_i\left(x^i_1\right), m_i\left(x^i_2\right), m_i\left(x^i_3\right), \ldots, m_i\left(x^i_{n_i-2}\right)\right]$, then

$$M_i(t_i) = \min\left\{m_i\left(x^i_1\right), \sum_{h=2}^{n_i-2} m_i\left(x^i_h\right) p^i_{h-1}(t_i)\right\} \quad (23)$$

**Step 5**: Return the process parameters $b_i(t_i)$, $C_i(t_i)$, and $M_i(t_i)$ to course P2.

In summary, **Step 1** determines the base matrix and the absorbing matrix. **Steps 2 to 4** are the core of the algorithm, determining process $i$'s completion probability, cost, and quality, respectively. **Step 5** returns the parameters.

## Numerical analysis

To illustrate the effectiveness of the proposed method, an example is given first. Then, the parameter tuning is analyzed. Finally, the comparison among GA–SA/CPM/Markov, GA/CPM/Markov, and SA/CPM/Markov is conducted for different problem size. The results show that the proposed GA–SA/CPM/Markov is effective.

### An example

An alliance intends to produce a new product-span truck. The network for processes is shown in Fig. 4. For the complete manufacture of the truck, from design to production, cost matrix and quality matrix of all processes are known and shown in Table 1.

For all processes, their completion time choices are shown in Table 2. Assume that the project's cost constraint $C_0$ is 19800 USD, time constraint $T_0$ is 59 days, and quality constraint is $M_{0i}$ $(i = 1, 2, \ldots N)$ which is obtained with the method in section "Problem description and the three-level model of dynamic risk-management planning".

As is shown, the project consists of 13 processes, each of which has several time choices. Therefore, there will be $\prod_{i=1}^{N} I_i = 155{,}520$ possible solutions.

### Parameters tuning

Options of the parameters of GA–SA are the main factors affecting the optimization and efficiency of the algorithm. The performances used in the parameters tuning are the "Best Rate", the rate to reach the best value. The "best value" stands for the best objective values achieved in 100 runs. The algo-

**Fig. 4** The network for the project

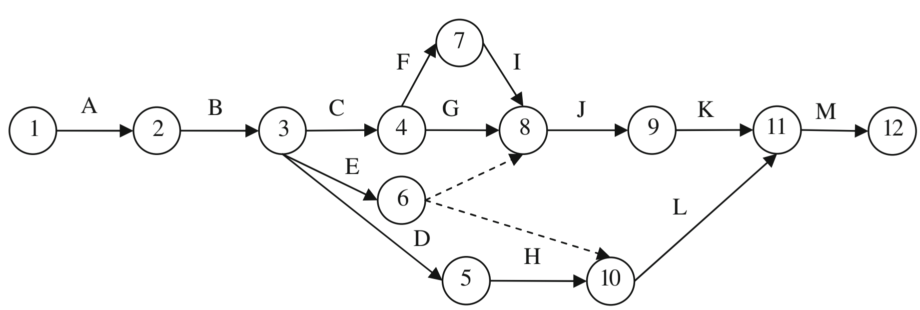

**Table 1** Details of the processes and their cost matrixes and quality matrixes

| Process name | Process code (successor processes) | Cost matrix | Quality matrix |
|---|---|---|---|
| Technology task book | A (B) | $f = [50, 250, 350, 300, 400]$ | $m = [0.9, 0.8, 0.7, 0.6, 0.3]$ |
| Technology design | B (C,E,D) | $f = [400, 2500, 2900, 2700, 3100]$ | $m = [0.9, 0.8, 0.6, 0.5, 0.2]$ |
| Machinery assembly design | C (F,G) | $f = [500, 1500, 2000]$ | $m = [0.9, 0.8, 0.4]$ |
| Electric apparatus assembly design | D (H) | $f = [125, 375, 500]$ | $m = [0.9, 0.8, 0.5]$ |
| Purchasing raw material and components | E (J,L) | $f = [500, 3600, 4500, 4000, 5000]$ | $m = [0.9, 0.8, 0.7, 0.4, 0.2]$ |
| Craft rules of machinery assembly | F (I) | $f = [200, 800, 1000]$ | $m = [0.9, 0.75, 0.4]$ |
| Parts design | G (J) | $f = [500, 1500, 2000]$ | $m = [0.9, 0.8, 0.4]$ |
| Craft rules of electric apparatus assembly | H (L) | $f = [40, 160, 200]$ | $m = [0.9, 0.8, 0.5]$ |
| Raw material quota | I (J) | $f = [125, 375, 500]$ | $m = [0.9, 0.8, 0.5]$ |
| Parts processing | J (K) | $f = [200, 600, 700, 800, 1000]$ | $m = [0.9, 0.8, 0.6, 0.4, 0.2]$ |
| Machinery assembly | K (M) | $f = [500, 3000, 3800, 4500, 5000]$ | $m = [0.9, 0.8, 0.6, 0.4, 0.2]$ |
| Electric apparatus assembly | L (M) | $f = [40, 160, 200]$ | $m = [0.9, 0.8, 0.5]$ |
| Testing | M | $f = [50, 200, 300]$ | $m = [0.9, 0.8, 0.6]$ |

**Table 2** Time choice of all processes

| Process code | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time choice | 2 | 8 | 8 | 4 | 5 | 5 | 8 | 1 | 2 | 5 | 16 | 1 | 1 |
| | 3 | 9 | 9 | 5 | 6 | 6 | 9 | 2 | 3 | 6 | 17 | 2 | 2 |
| | | 10 | 10 | | 7 | 7 | 10 | | | | 18 | | |
| | | | | | | | | | | | 19 | | |
| | | | | | | | | | | | 20 | | |

**Table 3** Effect of initial temperature on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 70 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 40 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 92 |
| 30 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 86 |
| 20 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 64 |

**Table 4** Effect of cooling rate on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.94 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.9 | 0.2 | 30 | 50 | 1 | 0.4 | 92 |
| 50 | 0.8 | 0.2 | 30 | 50 | 1 | 0.4 | 80 |
| 50 | 0.7 | 0.2 | 30 | 50 | 1 | 0.4 | 40 |

rithm was run 100 times with different random seeds for each parameter setting to test the random effect on the solution. Therefore, the parameters with highest "Best Rate" are better than others. The results are shown in Tables 3, 4, 5, 6, 7, 8 and 9.

Table 3 shows that the initial temperature $c_0$ must be large enough in order to attain the best solution. After $c_0$ rises to a certain level, there is no change in Best Rate. It is a waste of computation resource if temperature is kept on rising in order to reach the best solution. So 50 is a reasonable initial temperature.

Table 4 shows that the closer the cooling rate is to 1, the higher the Best Rate is.

Table 5 shows that the lower the stopping temperature is, the higher the Best Rate is. When it is lower than 0.4, the highest Best Rate is reached. Hence, the stopping temperature is set to 0.4 considering the speed of the algorithm.

Table 6 shows that the larger the maximum number of generation is, the higher the Best Rate is. When it is larger than 30, the highest Best Rate is reached. Hence, the maximum number of generation is set to 30 considering the speed of the algorithm.

Table 7 shows that the larger the population size is, the higher the Best Rate is. When it is larger than 50, the highest Best Rate is reached. Hence, the population size is set to 50 considering the speed of the algorithm.

**Table 5** Effect of stopping temperature on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.4 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.6 | 30 | 50 | 1 | 0.4 | 95 |
| 50 | 0.97 | 0.8 | 30 | 50 | 1 | 0.4 | 94 |
| 50 | 0.97 | 1 | 30 | 50 | 1 | 0.4 | 95 |

**Table 6** Effect of maximum number of generation on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 50 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 40 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 20 | 50 | 1 | 0.4 | 90 |
| 50 | 0.97 | 0.2 | 10 | 50 | 1 | 0.4 | 68 |

**Table 7** Effect of population size on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 30 | 60 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 40 | 1 | 0.4 | 92 |
| 50 | 0.97 | 0.2 | 30 | 30 | 1 | 0.4 | 78 |
| 50 | 0.97 | 0.2 | 30 | 20 | 1 | 0.4 | 54 |

Table 8 shows that the larger the crossover probability is, the higher the Best Rate is. When it is larger than 0.9, the highest Best Rate is reached.

Table 9 shows that the larger the mutation probability is, the higher the Best Rate is. When it is larger than 0.3, the highest Best Rate is reached.

Based on the analysis from Tables 3, 4, 5, 6, 7, 8 and 9, it can be found that the reasonable combination of parameters is:

$$c_0 = 50, r = 0.97, c_f = 0.2, GN = 30, NP = 50,$$
$$p_c = 1, p_m = 0.4$$

The best solution is:

$$t_1 = 3, t_2 = 9, t_3 = 10, t_4 = 5, t_5 = 7, t_6 = 7, t_7 = 10,$$
$$t_8 = 2, t_9 = 3, t_{10} = 6, t_{11} = 19, t_{12} = 2, t_{13} = 2$$

In summary, the total cost is 19409 USD, the completion time is 59 days, the quality value is 0.6695, and the maximum completion probability of the project is 0.5913.

**Table 8** Effect of crossover probability on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 0.9 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 0.8 | 0.4 | 95 |
| 50 | 0.97 | 0.2 | 30 | 50 | 0.7 | 0.4 | 93 |
| 50 | 0.97 | 0.2 | 30 | 50 | 0.6 | 0.4 | 93 |

**Table 9** Effect of mutation probability on Best Rate

| $c_0$ | $r$ | $c_f$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % |
|---|---|---|---|---|---|---|---|
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.4 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.3 | 98 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.2 | 95 |
| 50 | 0.97 | 0.2 | 30 | 50 | 1 | 0.1 | 95 |

**Table 10** Effect of problem size on Best Rate of GA–SA/CPM/Markov

| Size | $c_0$ | $c_f$ | $r$ | $GN$ | $NP$ | $p_c$ | $p_m$ | Best Rate % | CPU (s) |
|---|---|---|---|---|---|---|---|---|---|
| 155,520 | 50 | 0.2 | 0.97 | 30 | 50 | 1 | 0.4 | 98 | 5.9 |
| 23,245,229,000 | 50 | 0.2 | 0.97 | 30 | 50 | 1 | 0.4 | 92 | 10.45 |

## Comparison analysis

In order to show the proposed algorithm's effectiveness, a larger-size problem is given. For another example, there are 22 processes in a textile factory when producing colored woven cloth. There are 23,245,229,000 possible solutions. The comparison with the above example is shown in Table 10 where "Size" stands for the size of the space and "CPU" for the computation time of each running of CPU.

Table 10 shows that the Best Rate decreases when the problem size expands, but the rate tends to be stable as compared to the degree of size expansion. Generally speaking, the proposed algorithm has excellent potential.

The results of GA–SA/CPM/Markov are compared with those of GA/CPM/Markov (Table 11) and SA/CPM/Markov (Table 12), using the reasonable parameter setting for each algorithm.

Tables 10, 11 and 12 have shown that the Best Rate of GA–SA/CPM/Markov is more stable than GA/CPM/Markov and SA/CPM/Markov when the size of problem increases. Meanwhile, they have shown that GA–SA/CPM/Markov exhausts more time than the other two algorithms. However, the time increase is acceptable.

**Table 11** Effect of problem size on Best Rate of GA/CPM/Markov

| Size | *GN* | *NP* | $p_c$ | $p_m$ | Best Rate % | CPU (s) |
|---|---|---|---|---|---|---|
| 155,520 | 180 | 100 | 1 | 0.4 | 98 | 0.359 |
| 23,245,229,000 | 180 | 100 | 1 | 0.4 | 80 | 0.609 |

**Table 12** Effect of problem size on Best Rate of SA/CPM/Markov

| Size | $c_0$ | $c_f$ | $r$ | $M$ | Best Rate % | CPU (s) |
|---|---|---|---|---|---|---|
| 155,520 | 1,500 | 0.2 | 0.97 | 700 | 95 | 2.92 |
| 23,245,229,000 | 1,500 | 0.2 | 0.97 | 700 | 70 | 5.2 |

## Conclusion

With the economy's globalization and informationization, VEs play an important role in successfully coping with fast changing market and diversified customer demand. As well, VEs are also facing more risks. It is urgent to find a new risk management method to replace traditional ones. According to the characteristics of VEs in the form of project operation modes and with dynamic risks, we presented a three-level model of dynamic risk-management planning technologies. Non-linear integer programming is used to handle the objective and constraints of risk-management planning. Network analysis deals with link correlations among the processes. Markov process' principle explains the operation course's dynamic property. An algorithm of integrated GA–SA/CPM/Markov is designed to solve practical problems. Experimental results show that this method is efficient.

**Acknowledgments** This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant Nos. 71325002 and 61225012; the National Natural Science Foundation of China under Grant Nos. 71071028, 70931001 and 71021061; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No. 20120042130003; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110042110024; the Fundamental Research Funds for the Central Universities under Grant Nos. N110204003 and N120104001; the Fundamental Research Funds for State Key Laboratory of Synthetical Automation for Process Industries under Grant No.2013ZCX11. At the same time, the authors would like to thank the anonymous reviewers and the editor for constructive comments and suggestions.

## References

Bernus, P., & Nemes, L. (1999). Organisational design: Dynamically creating and sustaining integrated virtual enterprises. In H.-F. Chen, D.-Z. Cheng, & J.-F. Zhang (Eds.), *Proceedings of IFAC World Congress* (Vol. A, pp. 189–194). London: Elsevier.

Bier, V. M., Haimes, Y. Y., Lambert, J. H., Matalas, N. C., & Zimmerman, R. (1999). A survey of approaches for assessing and managing the risk of extremes. *Risk Analysis*, *19*(1), 83–94.

Boehm, B. W. (1991). Software risk management: Principles and practices. *IEEE Software*, *8*(1), 32–41.

Camarinha-Matos, L. M., & Pantoja-Lima, C. (2001). Cooperation coordination in virtual enterprises. *Journal of Intelligent Manufacturing*, *12*(2), 133–150.

Camarinha-Matos, L. M., & Afsarmanesh, H. (2003). Elements of a base VE infrastructure. *Computers in Industry*, *51*(2), 139–163.

Cardoso, H. L., & Oliveira, E. (2005). Virtual enterprise normative framework within electronic institutions. In *Engineering societies in the agents world V* (pp. 14–32). New York: Springer.

Carter, M. W., & Price, C. C. (2001). *Operations research: A practical introduction*. Boca Raton: CRC Press.

Chan, F. T. S., & Kumar, N. (2007). Global supplier development considering risk factors using fuzzy extended AHP-based approach. *Omega*, *35*(4), 417–431.

Chen, H. N., Zhu, Y. L., Hu, K. Y., & Li, X. H. (2010). Virtual enterprise risk management using artificial intelligence. In *Mathematical Problems in Engineering*. Hindawi Publishing Corporation. doi:10.1155/2010/572404.

Cowing, M. M., Paté-Cornell, M. E., & Glynn, P. W. (2004). Dynamic modeling of the tradeoff between productivity and safety in critical engineering systems. *Reliability Engineering and System Safety*, *86*(3), 269–284.

Goren, H. G., Tunali, S., & Jans, R. (2010). A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing*, *21*(4), 575–590.

Haimes, Y. Y. (2009). *Risk modeling, assessment, and management* (3rd ed.). New York: Wiley.

Hallikas, J., Virolainen, V. M., & Tuominen, M. (2002). Risk analysis and assessment in network environments: A dyadic case study. *International Journal of Production Economics*, *78*(1), 45–55.

Huang, M., Ip, W. H., Yang, H. M., Wang, X. W., & Lau, H. C. W. (2008). A fuzzy synthetic evaluation embedded tabu search for risk programming of virtual enterprises. *International Journal of Production Economics*, *116*(1), 104–114.

Huang, M., Lu, F. Q., Ching, W. K., & Siu, T. K. (2011). A Distributed decision making model for risk management of virtual enterprise. *Expert Systems with Applications*, *38*(10), 13208–13215.

Ibe, O. (2008). *Markov processes for stochastic modeling*. USA, Salt Lake City: Academic Press.

Ip, W. H., Huang, M., Yung, K. L., & Wang, D. W. (2003). Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise. *Computers and Operations Research*, *30*(2), 213–231.

Katzy, B. R., & Dissel, M. (2001). A tool set for building the virtual enterprise. *Journal of Intelligent Manufacturing*, *12*(2), 121–131.

Kierzkowski, Z. (2005). Towards virtual enterprises. *Human Factors and Ergonomics in Manufacturing*, *15*(1), 49–69.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Kliem, R. L., & Ludin, I. S. (1997). *Reducing project risk*. England: Gower Publishing Limited.

Li, Y., & Liao, X. (2007). Decision support for risk analysis on dynamic alliance. *Decision Support Systems*, *42*(4), 2043–2059.

Martinez, M. T., Fouletier, P., Park, K. H., & Favrel, J. (2001). Virtual enterprise-organization, evolution and control. *International Journal of Production Economics*, *74*(1), 225–238.

Michalewicz, Z. (1994). *Genetic algorithms + data structure = evolution programs* (2nd ed.). New York: Springer-Verlag.

Mun, J., Shin, M., Lee, K., & Jung, M. (2009). Manufacturing enterprise collaboration based on a goal-oriented fuzzy trust evaluation model in a virtual enterprise. *Computers and Industrial Engineering*, *56*(3), 888–901.

Park, K. H., & Favrel, J. (1999). Virtual enterprise-information system and networking solution. *Computers and Industrial Engineering*, *37*(1), 441–444.

Soares, A. L., Azevedo, A. L., & De Sousa, J. P. (2000). Distributed planning and control systems for the virtual enterprise: Organizational requirements and development life-cycle. *Journal of Intelligent Manufacturing*, *11*(3), 253–270.

Treleven, M., & Schweikhart, S. B. (1988). A risk/benefit analysis of sourcing strategies: Single versus multiple sourcing. *Journal of Operations Management*, *7*(4), 93–114.

Yen, C. H., Wong, D. S. H., & Jang, S. S. (2004). Solution of trim-loss problem by an integrated simulated annealing and ordinal optimization approach. *Journal of Intelligent Manufacturing*, *15*(5), 701–709.

Yu, V. F., Lin, S. W., Lee, W., & Ting, C. J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers and Industrial Engineering*, *58*(2), 288–299.

Zhang, D. Y., Cao, X. L., Wang, L. Y., & Zeng, Y. (2012). Mitigating the risk of information leakage in a two-level supply chain through optimal supplier selection. *Journal of Intelligent Manufacturing*, *23*(4), 1351–1364.