Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6916575, 17 pages
https://doi.org/10.1155/2017/6916575

Research Article

Simulated Annealing Genetic Algorithm Based Schedule Risk Management of IT Outsourcing Project

Fuqiang Lu, Hualing Bi, Min Huang, and Shupeng Duan

College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Hualing Bi; bihualing081@126.com

Received 10 April 2017; Revised 20 July 2017; Accepted 3 August 2017; Published 28 September 2017

Academic Editor: M. L. R. Varela

IT outsourcing is an effective way to enhance the core competitiveness for many enterprises. But the schedule risk of IT outsourcing project may cause enormous economic loss to enterprise. In this paper, the Distributed Decision Making (DDM) theory and the principal-agent theory are used to build a model for schedule risk management of IT outsourcing project. In addition, a hybrid algorithm combining simulated annealing (SA) and genetic algorithm (GA) is designed, namely, simulated annealing genetic algorithm (SAGA). The effect of the proposed model on the schedule risk management problem is analyzed in the simulation experiment. Meanwhile, the simulation results of the three algorithms GA, SA, and SAGA show that SAGA is the most superior one to the other two algorithms in terms of stability and convergence. Consequently, this paper provides the scientific quantitative proposal for the decision maker who needs to manage the schedule risk of IT outsourcing project.

1. Introduction

With the increasing development of information technology, IT outsourcing has been developing rapidly. It is currently being used as an important strategy by many companies to focus on the core competency, reduce cost, and increase profit. In Europe and other developed countries, either small businesses or large multinational companies always give the noncore business to external professional company [1–4]. According to Gartner, one of the leading information technology research firms, global spending for IT services was approximately $932 billion in 2013 and is expected to grow to $967 billion in 2014, a growth of 3.8% from 2013 [5].

Although IT outsourcing has many advantages including reducing cost and enhancing the core competence, there also exist some problems that need to be solved urgently, especially the problem of managing the schedule risk of IT outsourcing, which may bring about huge loss to company. Consequently, it is very vital to research how to manage the schedule risk of IT outsourcing.

Researchers have done a lot of related research [6–12]. But most methods and models proposed in the literature only discuss the risk management issues on project itself and ignore the cooperation between principal and agent and the distribution characteristics of the IT outsourcing activities. In recent years, principal-agent theory has been widely employed to solve the problem of risk management of IT outsourcing and good results have been achieved through these studies.

Earl et al. concluded some adverse consequences of IT outsourcing based on existing literatures [13–16]. Then, he argued that the risk of IT outsourcing came from enterprises, agents, and the process of IT activities and proposed corresponding risk management measures based on principal-agent theory [17–19]. Bahli and Rivard proposed a scenario-based conceptualization of the IT outsourcing risk and applied it to the specific context of IT outsourcing using transaction cost and agency theory [20]. Osei-Bryson and Ngwenyama offered a method and some mathematical models for analyzing risks and constructing incentive contracts for information system outsourcing [21]. Sanfa et al. analyzed risk factors of producer services outsourcing from the perspective of engineering and afforded managers a theoretic method to manage outsourcing risks by designing the incentive and monitoring mechanism of the producer services outsourcing contract [22]. Xianli et al. present the idea of applying DDM (distributed decision making) to the risk management of virtual enterprises and design incentive and punishment mechanism in the principal-agent model [23].

In this paper, we build a two-level principal-agent model combined with reward and punishment mechanism for schedule risk management of IT outsourcing project based on the Distributed Decision Making (DDM) theory and principal-agent theory [24–26]. According to the feature of problem, the SAGA is designed to solve the proposed model and the optimal plan of managing schedule risk is given based on the simulation analysis. The purpose of this paper is to provide crucial decision support for the people who need to manage the schedule risk of IT outsourcing project.

The remainder of this paper is structured as follows. Section 1 presents the schedule risk management model of IT outsourcing project. In Section 2, the design of algorithm is given. In addition, numerical examples and results analyzed are depicted in Section 3. Finally, conclusion is given in Section 4.

2. Schedule Risk Management Model of IT Outsourcing Project

2.1. Problem Description. For IT outsourcing, principal divides a whole project into some serial subprojects in the IT developing process, as shown in Figure 1. The definition of serial subprojects is that subproject i $(i = 2, \ldots, I)$ is performed after completion of subproject $i - 1$.

The schedule risk is reflected in two aspects of duration and risk loss. Each subproject has an initial duration and initial risk loss. In order to effectively manage the schedule risk, the reward and punishment mechanism is added in outsourcing contract; that is, if the project is completed in advance, the agent is rewarded; otherwise the agent is punished. Each subproject will be contracted with different agents, and a typical principal-agent relationship between principal and agent will be generated. For the relationship between principal and agents, see Figure 2.

The optimal solution of top-level model is the optimal combination of risk management capital, and the optimal solution of base-level model is the optimal combination of risk management measure of subproject. In the decision making process, the principal transfers risk management capital to the predicted base-level model. The optimal solution of top-level model is obtained based on the goal of maximizing the profit of the principal and the information returned from the predicted base-level model. Then, the optimal solution of top-level model is transferred to the real base-level model. Under the constraint of risk management capital, the agents obtain the best control measure combination of the subproject according to the goal of maximizing the profit. The information exchange process between the principal and the agents is shown in Figure 3.

For the IT development, the duration of the subproject is determined by the duration of the activities on the critical path. Hence we only consider the schedule of the activities on the critical path. Figure 4 shows the network diagram of subproject I, in which the critical path is 1–3-4–6-7-8. So agent i only allocates risk management capital to activities 1, 2, 3, 6, 8, and 9.

2.2. Assumptions

(1) Subprojects are serial relation in the IT developing process.

(2) The change of completion probability or duration of subproject will not have an effect on other subprojects; that is, subprojects are independent of each other.

(3) The critical path of subproject will not become noncritical path under the influence of risk management capital.

(4) The duration and risk loss of project only are affected by risk management capital.

(5) Due to the information asymmetry, the risk loss per unit of subproject is clear to the agents, but the principal only masters its distribution function.

2.3. The Two-Level Principal-Agent Model. Based on the DDM theory and principal-agent theory, a two-level schedule risk management model of IT outsourcing project is built [27, 28]. In the top-level, the decision maker is the principal who determines how to allocate the risk management capital among agents. The objective of top-level is to maximize the profit of principal, and the reward and punishment mechanism is introduced into the model. In the base-level, the decision maker is the agents who determine the best combination of risk management measure of subproject. The objective of base-level is to maximize the agent's benefit.

2.3.1. Top-Level

Variable Definition

x_i: risk management capital of subproject i

$\hat{y}_i$: the predicted combination of risk management measure of subproject i

b_i: the agent's profit sharing coefficient of subproject i

I: the number of agents or subprojects

$t_i(\hat{y}_i)$: predicted duration of subproject i

T_i^0: planned duration of subproject i

$\Delta L_i(\hat{y}_i)$: predicted saved risk loss of subproject i

$B_i(x_i, \hat{y}_i)$: predicted profit of agent i

$X^{\max}$: risk management capital budget

$h_i(\hat{y}_i)$: predicted reward and punishment function of subproject i

AL_i: the aspiration level of agent i

e_i: additional profit per time unit of subproject i.

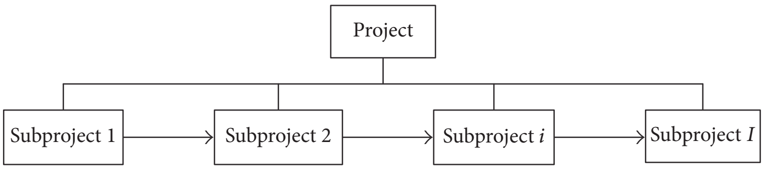


Figure 1: The diagram of project construction.

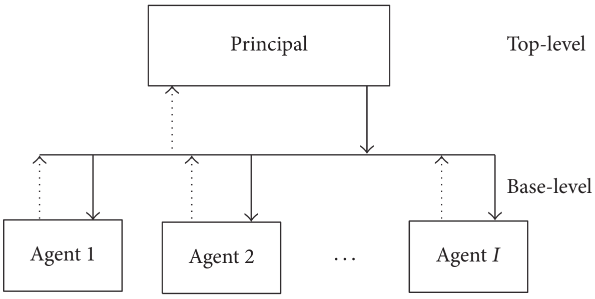


Figure 2: The relationship between principal and agents.

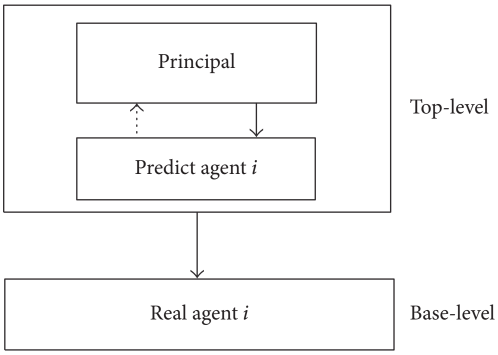


Figure 3: The information exchange process between the principal and the agents.

$$\max \quad \sum_{i=1}^{I} b_i \Delta L_i \left(\hat{y}_i\right) - x_i + h_i \left(\hat{y}_i\right) - e_i \left(t_i \left(\hat{y}_i\right) - T_i^0\right)^- \tag{1}$$

$$\text{s.t.} \quad B_i \left(x_i, \hat{y}_i\right) \geq \mathrm{AL}_i \tag{2}$$

$$B_i \left(x_i, \hat{y}_i^*\right) \geq B_i \left(x_i, \hat{y}_i\right) \tag{3}$$

$$\sum_{i=1}^{I} x_i \leq X^{\max} \tag{4}$$

$$x_i \in N^+. \tag{5}$$

The objective of top-level shown in formula (1) is to maximize the profit of principal; the reward and punishment mechanism is fulfilled by item $h_i(\hat{y}_i)$. The operation $(x)^-$ is defined as

$$(x)^- = \begin{cases} x, & x < 0, \\ 0, & \text{else}. \end{cases} \tag{6}$$

Formula (2) indicates participation constraint; formula (3) indicates incentive compatibility constraint; formula (4) indicates risk management capital constraint; formula (5) indicates that the risk management capital x_i is a natural integer, which is the decision variable in the top-level model.

2.3.2. Predicted Base-Level. In the predicted base-level, the decision makers are the agents, and there are I agents. Take agent i, for example.

Variable Definition

Y_{ij}: the number of management measures of activity j

J_i: the number of activities of subproject i

L_i: initial risk loss of subproject i

ξ_i: predicted risk loss per unit of subproject i

q_i: reward and punishment standard of subproject i

$c_i(t_i(\hat{y}_i))$: predicted cost of subproject i

$d_i(\hat{y}_i)$: predicted invested risk management capital of subproject i

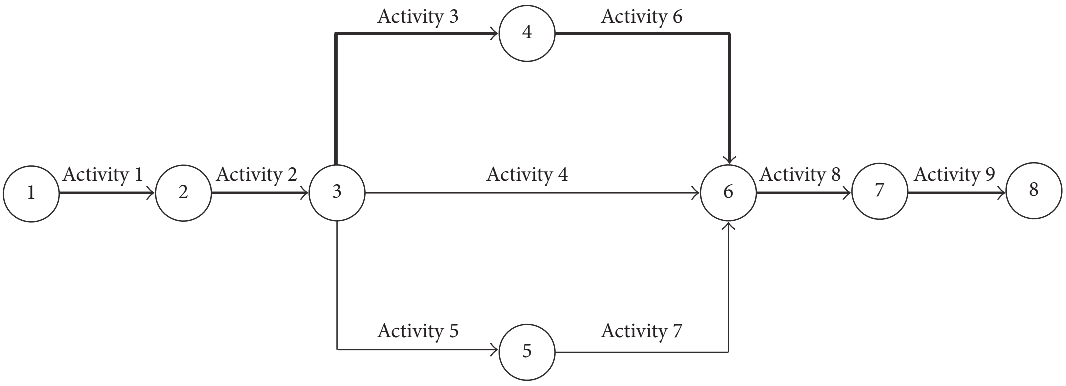


Figure 4: The network diagram of subproject i.

α_i: confidence level

$C_i^{\max}$: the upper bound of cost of subproject i.

$$B_i(x_i, \hat{y}_i) = \max \quad B_i \tag{7}$$

$$\text{s.t.} \quad \text{pr}\left\{b_i \Delta L_i(\hat{y}_i) - c_i(t_i(\hat{y}_i)) + x_i - d_i(\hat{y}_i) - h_i(\hat{y}_i) \geq B_i\right\} \geq \alpha_i \tag{8}$$

$$h_i(\hat{y}_i) = q_i\left(t_i(\hat{y}_i) - T_i^0\right) \tag{9}$$

$$\Delta L_i(\hat{y}_i) = L_i - a_i\left(t_i(\hat{y}_i) - T_i^0\right)^+ \tag{10}$$

$$d_i(\hat{y}_i) \leq x_i \tag{11}$$

$$c_i(t_i(\hat{y}_i)) \leq C_i^{\max} \tag{12}$$

$$\hat{y}_i = \left(\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iJ_i}\right) \tag{13}$$

$$\hat{y}_{ij} \in \left\{1, 2, \ldots, Y_{ij}\right\}. \tag{14}$$

Formula (7) indicates that the objective of predicted base-level is to maximize agent's benefit. Formula (8) indicates chance constraint. Formula (9) indicates the reward and penalty function based on the duration. Formula (10) indicates the saved risk loss of subproject; the operation $(x)^+$ is defined as

$$(x)^+ = \begin{cases} x, & x > 0, \\ 0, & \text{else.} \end{cases} \tag{15}$$

Formula (11) indicates that the sum of used risk management capital is not greater than the risk management capital which is allocated to subproject; formula (13) represents a set of the predicted base-level variables; formula (14) represents the value range of $\hat{y}_{ij}$ that is the decision variable in the predicted base-level model.

2.3.3. Real Base-Level.

2.3.3. Real Base-Level. In the real base-level, the decision makers are the agents, and there are I agents. Take agent i, for example.

Variable Definition

y_i: the actual combination of risk management measure of subproject i

a_i: risk loss per unit of subproject i

$\Delta L_i(y_i)$: actual saved risk loss of subproject i

$t_i(y_i)$: actual duration of subproject i

$h_i(y_i)$: actual reward and penalty function of subproject i

$d_i(y_i)$: actual invested risk management capital of subproject i

$c_i(t_i(y_i))$: actual cost of subproject i.

$$B_i\left(x_i^*, y_i\right) = \max \quad b_i \Delta L_i\left(y_i\right) - c_i\left(t_i\left(y_i\right)\right) x_i^* - d_i\left(y_i\right) - h_i\left(y_i\right) \tag{16}$$

$$\text{s.t.} \quad h_i\left(y_i\right) = q_i\left(t_i\left(y_i\right) - T_i^0\right) \tag{17}$$

$$\Delta L_i\left(y_i\right) = L_i - a_i\left(t_i\left(y_i\right) - T_i^0\right)^+ \tag{18}$$

$$d_i\left(y_i\right) \leq x_i^* \tag{19}$$

$$c_i\left(t_i\left(y_i\right)\right) \leq C_i^{\max} \tag{20}$$

$$y_i = \left(y_{i1}, y_{i2}, \ldots, y_{iJ_i}\right) \tag{21}$$

$$y_{ij} \in \left\{1, 2, \ldots, Y_{ij}\right\}. \tag{22}$$

Formula (16) indicates that the objective of real base-level is to maximize agent's benefit; formula (17) indicates the reward and penalty function based on the duration; formula (18) indicates the saved risk loss of subproject; formula (19) indicates that the sum of used risk management capital is not greater than the risk management capital which is allocated to subproject; formula (20) represents the cost that should not be beyond the upper bound of cost for subproject *i*; formula (21) represents a set of the real base-level variables; formula (22) represents the value range of y_{ij} that is the decision variable in the real base-level model.

3. Algorithm Design

The top-level model is an integer programming problem, and the base-level model (including the predicted base-level model) is a combinatorial optimization problem. The whole problem is a NP hard problem, because the base-level is embedded in the top-level. So, we use genetic algorithm (GA) to solve the problem in this paper. It is well known that GA that was first introduced by Holland is very effective for solving combinatorial optimization problems. For example, GA has been successfully applied in solving traveling salesman problem, knapsack problem, bin packing problem, and so on. However, the disadvantage of GA is that the local search capability is not strong [29–33]. Simulated annealing (SA) is a general random search algorithm, which is an extension of the local search algorithm [34–37]. Considering the strong local search capability of SA, we designed a hybrid algorithm named simulated annealing genetic algorithm (SAGA) by combining simulated SA with GA.

The overall thought of SAGA is simple. Firstly, some initial solutions of GA are generated randomly. After a period of iteration, some superior solutions are produced. Then, sort the corresponding fitness value of these superior solutions in descending order. And then select the solutions in top 10% as the initial solutions of SA. We try to find the best solution of the proposed problem around these superior solutions. For the random variables of the predicted base-level model, we embed Monte Carlo Simulation in SAGA.

3.1. Encoding Scheme. In top-level, each chromosome represented by real number is a combination of risk management capital and the length of chromosome represents the number of agents. The top-level encoding scheme of SAGA is shown in Figure 5. We can see that there are 4 agents, and $580.2 is allocated to agent 1; $600.9 is allocated to agent 2; the rest can be done in the same manner.

In base-level, each chromosome represented by real number is a combination of risk management capital and the length of chromosome represents the number of the activities on the critical path. Take the base-level encoding scheme shown in Figure 6 as example; it can be seen that there are 5 activities on the critical path, and measure 2 is used to manage the risk of activity 1, measure 4 is used to manage the risk of activity 2, and so on.

3.2. Population Initialization. Initial population is generated randomly. Punishment strategy is adopted to deal with the constraints, so we do not have to judge whether the initial solution meets the constraint conditions.

3.3. Fitness Function. Considering the proposed optimization problems with constraints, we set up a fitness function with punishment term to evaluate individuals. The top-level fitness function is given as

$$\begin{aligned} F_T = f_T - \sum_{i=1}^{I}\left(\alpha_i\left(B_i\left(x_i, \hat{y}_i\right) - \mathrm{AL}_i\right)\right) \\ - \beta\left(\sum_{i=1}^{I} x_i - X^{\max}\right), \end{aligned} \tag{23}$$

where

$$f_T = \sum_{i=1}^{I} b_i \Delta L_i\left(\hat{y}_i\right) - x_i + h_i\left(\hat{y}_i\right) - e_i\left(t_i\left(\hat{y}_i\right) - T_i^0\right)^-. \tag{24}$$

Function (24) is top-level objective function; α_i and β are punishment coefficients, respectively.

Agent	1	2	3	4
Capital	580.2	600.9	643.7	576

Figure 5: The top-level encoding scheme of SAGA.

Activity	1	2	3	4	5
Measure	2	4	2	3	1

Figure 6: The base-level encoding scheme of SAGA.

The base-level fitness function is given as

$$\begin{aligned} F_{Bi} = f_{Bi} - \gamma_i \left(d_i\left(y_i\right) - x_i^*\right)^+ \\ - \delta_i \left(c_i\left(t_i\left(y_i\right)\right) - C_i^{\max}\right)^+, \end{aligned} \qquad (25)$$

where

$$f_{Bi} = b_i \Delta L_i\left(y_i\right) - c_i\left(t_i\left(y_i\right)\right) + x_i^* - d_i\left(y_i\right) - h_i\left(y_i\right). \qquad (26)$$

Function (25) is base-level objective function; γ_i and δ_i are punishment coefficients, respectively.

The Monte Carlo Simulation method is used to deal with the random variable [37–40]. The process of calculating the fitness of the predicted base-level model by Monte Carlo Simulation is shown as follows.

Step 1. Set Q' is equal to $[\alpha_i Q]$; Q is sampling number.

Step 2. Samples $\xi_i^1, \xi_i^2, \ldots, \xi_i^Q$ are generated by normal distribution function $N(a_i, 0.01)$.

Step 3. Calculate F_{Bi}^t that is the fitness value of the predicted base-level model by formula (26), $t = 1, 2, \ldots, Q$.

Step 4. The Q'th biggest element of $\{F_{Bi}^t, F_{Bi}^t, \ldots, F_{Bi}^t\}$ can be used as the fitness value of the ith predicted base-level model, which can be seen by the law of large numbers.

3.4. Selection. This paper takes proportional selection strategy [41, 42]. For each individual, the probability of being selected is the proportion of its fitness to the sum of all individuals' fitness. Then, the probability of selected individual I is given by

$$P_i = \frac{F_i}{\sum_{i=1}^{NP} F_i}, \qquad (27)$$

where F_i is the fitness of individual i and NP is the population size. Here, we adopt well-known Roulette Wheel scheme. In order to prevent the best individual in each generation from being destroyed, elite-preservation strategy is also used. That is to say, the best individual of each generation directly becomes one of individuals in the next generation without crossover and mutation operation.

3.5. Crossover. Double-point crossover is adopted in this paper, which is beneficial for keeping excellent individual. Figure 7 shows the example of the double-point crossover operator, where $P_1 = (1, 3, 3, 2, 4)$ and $P_2 = (2, 2, 5, 1, 3)$ are parent chromosomes. So, the generated children chromosomes are $C_1 = (1, 2, 5, 1, 4)$ and $C_2 = (2, 3, 3, 2, 3)$.

3.6. Mutation. Reversal mutation is adopted in this paper [35]. Under the condition of satisfying the mutation rate, randomly select two points in the parent and sort the genes between these two points in reverse order. The reversal mutation operator of SAGA is shown in Figure 8. Parent chromosome $P = (1, 4, 2, 3, 1)$ is selected for mutation operation. And the generated children chromosome is $P' = (1, 2, 4, 3, 1)$.

3.7. Neighborhood Definition. The bit x_i is selected from the current state $x = [x_1, x_2, \ldots, x_m]$, and the value of x_i is changed in the range of its value. So, a neighborhood solution is generated.

3.8. Neighborhood Movement. The Metropolis criterion is adopted in this paper. If the objective value of neighborhood solution is smaller than the current solution's objective value, the current solution is replaced by the neighborhood solution. Otherwise, the current solution moves according to a certain probability.

3.9. Thermal Equilibrium. Thermal equilibrium is achieved when the preset number of internal loop is reached.

3.10. Cooling Rule. Reduce T_k by multiplying a number r, which is in the range $[0, 1]$ and close to 1. Cooling rule is shown as

$$T_{k+1} = T_k, \qquad (28)$$

where T_k is the current temperature, T_0 is initial temperature, k is the iterative index, and $r \in (0.95, 0.99)$.

3.11. Procedure of SAGA. The flow chart of SAGA is shown in Figure 9. The process of the SAGA is made up of two parts, GA and SA. In Figure 9, the left part is the process of GA; GA

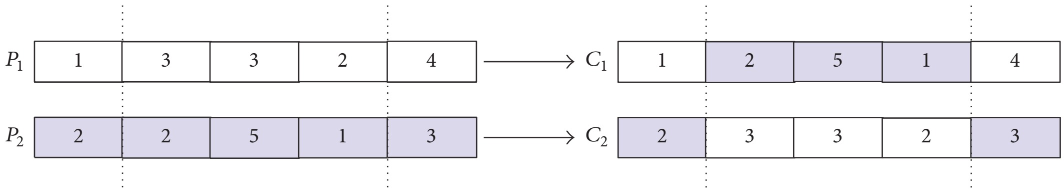


Figure 7: The double-point crossover operator of SAGA.

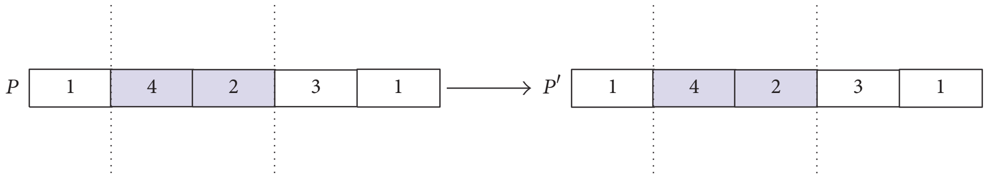


Figure 8: The reversal variation operator of SAGA.

is used firstly to find a good solution of the problem. To cover the defect of GA on the local area search, SA is used to find an optimal solution around the "good solution" find by GA. The process of SA is shown on the right part of the figure. After the local search of SA, the proposed algorithm ended.

4. Numerical Examples

4.1. Example 1. Take the IT outsourcing project including 1 principal and 1 agent ($I = 1$) for an example. The principal outsources the whole project to the agent. Schedule risk is reflected by duration and risk loss of project. It is assumed that if the project is completed before planned duration, there will be no risk loss. On the contrary, the project will lose \$80 per day; namely, $a_1 = 80$. Due to the information asymmetry, the agent is clear for parameter a_1, but the principal only knows its distribution function, which is approximated as $N(80, 0.01)$. The planned duration of project T_1^0 is 400, the initial duration of project T_1 is 500, and the initial duration of activities t_{1j}^0 is $\{55, 60, 65, 70, 75, 80, 95\}$. The profit sharing coefficient of principal b_1 is 0.51, the expected profit of agent AL_1 is 2000, and confidence level α is 0.9.

There is an initial schedule risk, which can be seen from the parameters of duration. The principal prepares \$900 to manage schedule risk, $X^{\max} = 900$. The project consists of 7 activities, $J_1 = 7$. There are 10 available risk management measures for each activity, $Y_{1j} = 10$, $j \in \{1, 2, \ldots, 7\}$. It is assumed that the agent must choose a numbered measure for each activity and that the selected numbered measure can be selected multiple times.

Risk management measure is the only variable that affects the duration and risk loss of project, which can be seen from the proposed model. It is assumed that the impact of risk management measure on the project is enhanced with the increase of the number of measures. For example, for an activity, the implementation of measure 7 will result in a shorter duration and a smaller risk loss than the implementation of measure 2. So, the monotonic decreasing function is used to represent the duration function that is shown in formula (29), and risk loss function is shown in formula (30), respectively,

$$t_1(y_1) = \sum_{j=1}^{J_1} t_{1j}^0 \exp\left(-y_{1j}\sigma_{1j}\right), \tag{29}$$

$$L_1(y_1) = a_1\left(t_1(\hat{y}_1) - T_1^0\right)^+, \tag{30}$$

where σ_{1j} is control parameter, which is used to represent that a management measure has different effects on different activities. The saved risk loss of project is shown as follows:

$$\Delta L_1(y_1) = a_1\left(T_1 - T_1^0\right) - L_1(y_1). \tag{31}$$

If the project can be completed before the planned duration, the principal will obtain additional profit. It is assumed that the additional profit e_1 is \$25 per time unit. So, the reward and punishment function about duration is shown in formula (32), which is designed by the principal

$$h_1(y_1) = q_1\left(t_1(y_1) - T_1^0\right). \tag{32}$$

It is assumed that the reward and punishment parameter q_1 is \$21. If the subproject is completed in advance, the agent can get \$21 per day; otherwise, he will lose \$21 per day.

4.1.1. The Parameters of SAGA. The parameters of SAGA mainly include population size (NP), maximum generations (NG), crossover rate (Pc), mutation rate (Pm), initial temperature (Ti), termination temperature (Ts), and internal loop number (Ni). The parameters have a significant impact on the performance of algorithm. For example, big population size may lead to slower convergence speed but can avoid suboptimal solution. Small population size may lead to premature but can ensure the quick speed of convergence.

In this paper, NG is determined by

$$NG = \frac{1500}{NP}. \tag{33}$$

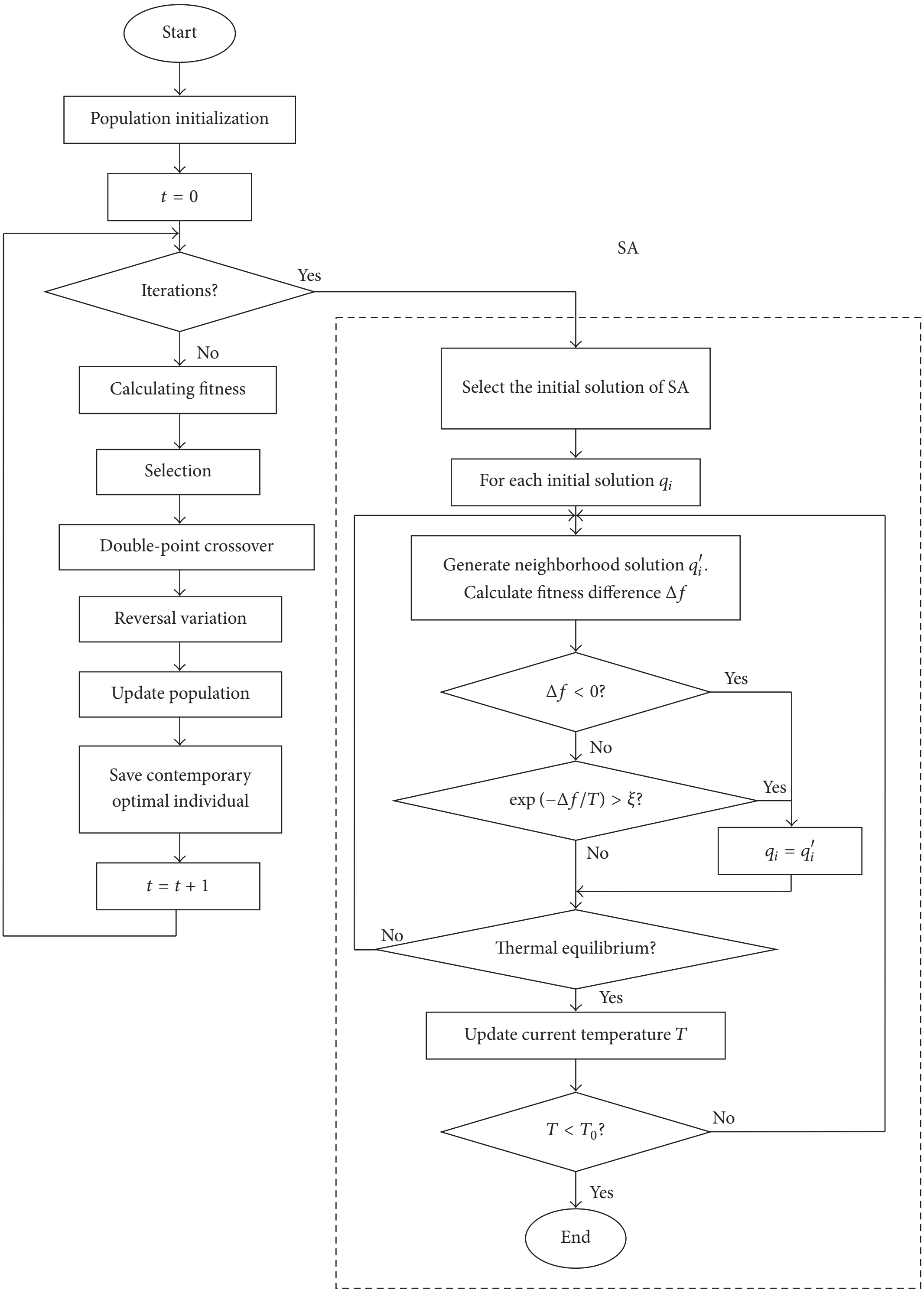

Start
Population initialization
$t = 0$
Iterations?
Yes
No
Calculating fitness
Selection
Double-point crossover
Reversal variation
Update population
Save contemporary optimal individual
$t = t + 1$
SA
Select the initial solution of SA
For each initial solution q_i
Generate neighborhood solution q_i'.
Calculate fitness difference Δf
$\Delta f < 0$?
Yes
No
$\exp(-\Delta f/T) > \xi$?
Yes
No
$q_i = q_i'$
Thermal equilibrium?
No
Yes
Update current temperature T
$T < T_0$?
No
Yes
End

Figure 9: The flow chart of SAGA.

Generally, Ts is a smaller positive number; let $Ts = 1$ here. So, we only need to test the other 5 parameters with the following method.

(1) Provide two values for each parameter. One is relatively small, and the other is relatively big.

(2) For each combination of parameter, run algorithm 20 times. There are 2^5 combinations.

(3) The final parameter combination is the one with best average fitness value.

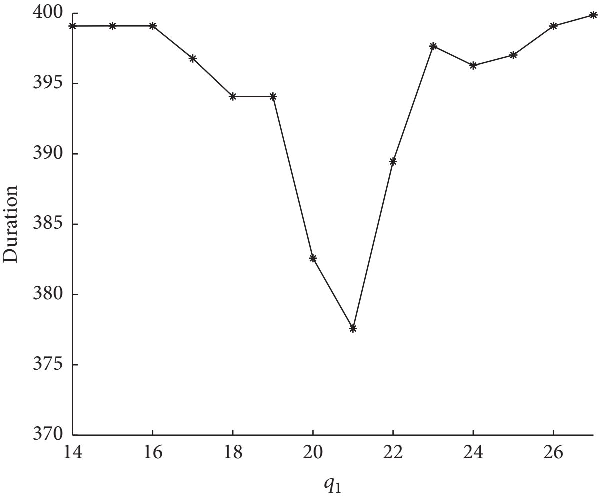


Figure 10: The influence of q_1 on duration.

The testing process is shown in Table 1. The final combination is $\{30, 50, 0.6, 0.1, 100, 10\}$.

4.1.2. Simulation Results. The simulation results of example 1 are shown in Table 2. For real results, agent is allocated \$597 and he selects measures 2 1 10 1 10 10 10 to manage risk. After that, the duration of project is reduced from 500 days to 377.41 days; that is to say, the project is completed in advance. For index duration and risk loss, the schedule risk of IT outsourcing project has been well managed.

It can be seen that the predicted results and real results are different, such as profit of principal, because there is information asymmetry between principal and agent, and the principal's prediction is not completely correct in the real situation, which always plays an important role in the process of schedule risk management. If there is no prediction, the whole project would be completely out of control; the effect of schedule risk management will be worse than it is now.

4.1.3. Model Comparison. Model I refers to the proposed model in Section 2.3; Model II is the one in which there is no reward and punishment item. Model I is Model II, when the parameter $q_1 = 0$. For the simulation results in Table 3, from index 3 to 8, Model I is better than Model II. Therefore, it is significant to join the reward and punishment mechanism in the model.

4.1.4. The Parameter of Reward and Punishment Mechanism. If parameter q_1 is not set reasonably, the reward and punishment will not really play its role. Therefore, the effect of the parameter q_1 is tested on the proposed model. The test results are shown in Table 4. The influence of q_1 on the duration, the profit of agent, and the profit of principal are shown in Figures 10–12, respectively.

We can get some information from Table 4 and Figures 10–12. When $q_1 \leq c_1 = 16$ where c_1 is additional cost per day due to duration reduction, total risk loss is saved, but duration is not significantly reduced compared to the planned duration. The reason for the above phenomenon is that the agent will not make efforts to shorten duration, if the reward is not greater than the cost because of duration reduction. When $q_1 \geq e_1 = 25$, the reward of agent is greater than the additional profit of principal due to shorting duration, resulting in the decline of principal's profit. Obviously, this is not acceptable to the principal. So, we must determine the specific value of q_1 in the interval (c_1, e_1) based on the 4 indexes: duration, saved risk loss, profit of agent, and profit of principal. Considering managing the schedule risk of IT outsourcing project, we set q_1 to be 21. Finally, the way for setting the reward and punishment mechanism's parameter is summarized as follows.

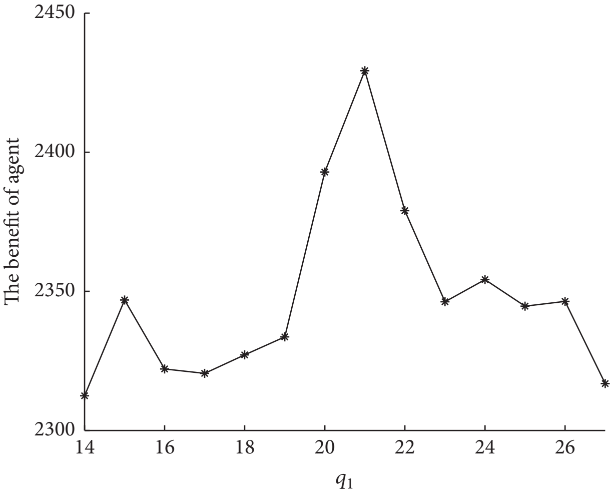


Figure 11: The influence of q_1 on the profit of agent.

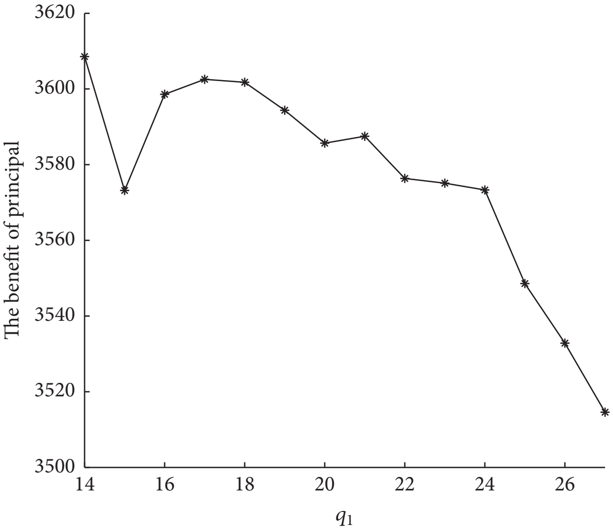


Figure 12: The influence of q_1 on the profit of principal.

(1) Set the two values c and e, $c < e$.

(2) Determine the specific value of q_1 in the interval $[c, e]$ based on the 4 indexes: duration, recovered risk loss, profit of agent, and profit of principal.

Table 1: The parameters of SAGA.

NP	*NG*	*Pc*	*Pm*	*Ti*	*Ni*	Fitness
30	50	0.9	0.1	130	20	2417.8
30	50	0.9	0.1	130	10	2352.3
30	50	0.9	0.1	100	20	2344.9
30	50	0.9	0.1	100	10	2327.5
30	50	0.9	0.01	130	20	2417.3
30	50	0.9	0.01	130	10	2407.6
30	50	0.9	0.01	100	20	2327.9
30	50	0.9	0.01	100	10	2423.4
30	50	0.6	0.1	130	20	2334.8
30	50	0.6	0.1	130	10	2338.3
30	50	0.6	0.1	100	20	2348.7
30	**50**	**0.6**	**0.1**	**100**	**10**	**2441.1**
30	50	0.6	0.01	130	20	2417.4
30	50	0.6	0.01	130	10	2339.2
30	50	0.6	0.01	100	20	2326.3
30	50	0.6	0.01	100	10	2404.1
50	30	0.9	0.1	130	20	2319.5
50	30	0.9	0.1	130	10	2408.2
50	30	0.9	0.1	100	20	2328.6
50	30	0.9	0.1	100	10	2328.2
50	30	0.9	0.01	130	20	2402.0
50	30	0.9	0.01	130	10	2316.8
50	30	0.9	0.01	100	20	2322.8
50	30	0.9	0.01	100	10	2317.0
50	30	0.6	0.1	130	20	2394.7
50	30	0.6	0.1	130	10	2396.3
50	30	0.6	0.1	100	20	2324.5
50	30	0.6	0.1	100	10	2326.3
50	30	0.6	0.01	130	20	2361.7
50	30	0.6	0.01	130	10	2319.3
50	30	0.6	0.01	100	20	2367.9
50	30	0.6	0.01	100	10	2321.6

Table 2: The simulation results of example 1.

Index	Predicted results	Real results
Allocated capital	597	597
Measure combination	1 10 1 1 9 10 10	2 1 10 1 10 10 10
Duration	380.52	377.41
Finish the project before planned duration	Y	Y
Saved risk loss	7985.6	8000
Save total risk loss?	Y	Y
Profit of agent	2430.69	2440.50
Profit of principal	3565.15	3564.98

4.1.5. Performance Comparison of Algorithms. In this paper, we use three algorithms, SA, GA, and SAGA to carry out the simulation calculation for the proposed model. The simulation results are compared in Table 5. Focusing on indexes, duration, finishing the project before planned duration, saved risk loss, total risk loss saved, profit of agent, and profit of principal, the conclusion can be drawn that SAGA is better than GA and SA

The convergence process of 3 algorithms in example 1 is shown in Figure 13. For convergent speed, SA achieves its

Table 3: The model comparison of example 1.

Index	Mode I	Model II
Allocated capital	597	502
Measure combination	2 1 10 1 10 10 10	2 1 1 9 1 10 10
Duration	377.41	400.98
Finish the project before planned duration	Y	N
Saved risk loss	8000	7921.60
Save total risk loss?	Y	N
Profit of agent	2440.50	2306.76
Profit of principal	3564.98	3532.84

Table 4: The effect of q_1 on the model.

q_1	Duration	Saved risk loss	Profit of agent	Profit of principal
14	399.09	8000	2312.44	3608.55
15	399.10	8000	2346.89	3573.18
16	399.10	8000	2322.08	3598.58
17	396.78	8000	2320.52	3602.54
18	394.08	8000	2327.17	3601.78
19	394.08	8000	2333.63	3594.37
20	382.58	8000	2392.89	3585.67
21	**377.58**	**8000**	**2429.31**	**3587.50**
22	389.46	8000	2379.02	3576.35
23	397.96	8000	2346.25	3575.10
24	396.28	8000	2354.21	3573.32
25	397.03	8000	2344.66	3548.59
26	399.09	8000	2346.40	3532.84
27	399.88	8000	2316.86	3514.58

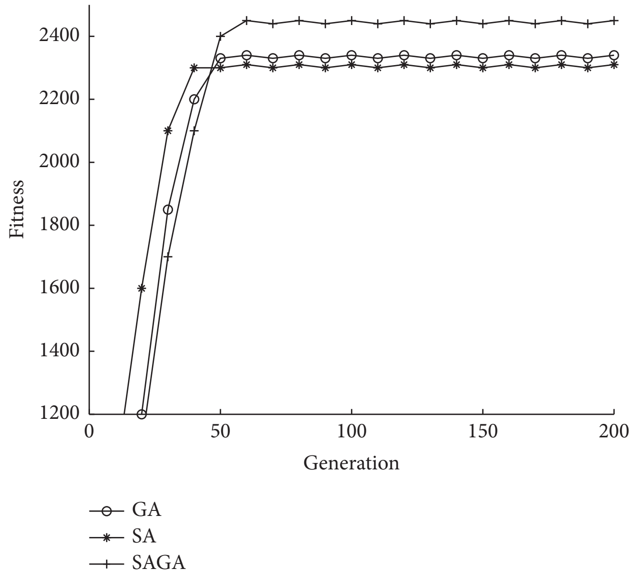


Figure 13: The convergence process of the three algorithms in example 1.

convergence after about 40 iterations; GA and SAGA need about 50 iterations. So, the convergence speed of the three algorithms is near. However, considering the fitness from the three algorithms, it is easy to see that SAGA finds the best solution, among the three algorithms. Therefore, SAGA is the best choice for the problem in example 1.

4.2. Example 2, 3, and 4. In this section, further examples 2, 3, and 4 are given in this section to test the performance of SAGA. The values of the main parameters for models in the three examples are shown in Table 6, and the values of others parameters are the same to example 1.

4.2.1. The Parameters of SAGA. According to the method of parameters setting in example 1, the parameters combination of SAGA for example 2, 3, and 4 is shown in Table 7.

4.2.2. Simulation Results. The simulation results of examples 2 and 3 are shown in Tables 8 and 9, respectively. The comprehensive results of example 4 are not shown in this section, for its large amount of data, but a part of selected results is used to analyze the algorithm in the next section.

For real results of example 3, the principal allocates $682, $579, $579, $649, $609, $579, $689, $689, $617, and $646 to 10 agents, respectively. The agents select the measure combination shown in table to manage risk. After that, the duration of projects is reduced from 500 days to 391.4, 400.1,

Table 5: The comparison of three algorithms.

Index	GA	SA	SAGA
Allocated capital	542	556	597
Measure combination	1 2 9 1 5 10 7	1 2 1 1 9 10 10	2 1 10 1 10 10 10
Duration	399.76	400.07	377.41
Finish the project before planned duration?	Y	N	Y
Saved risk loss	8000	7943.24	8000
Save total risk loss?	Y	N	Y
Profit of agent	2325.04	2313.29	2440.50
Profit of principal	3540.89	3515.23	3564.98

Table 6: The model parameters of examples 2, 3, and 4.

Example	Principal's number	Agent's number	Activity's number	Measure's number	Risk management capital
2	1	5	7	10	3500
3	1	10	7	10	7000
4	1	25	7	10	17500

400.1, 384.7, 397.1, 397.1, 384.7, 368.5, 386.4, and 379. days, and total risk loss is saved except for subprojects 2 and 3. For the two indexes of duration and risk loss, the schedule risk of IT outsourcing project has been well managed.

We can see that the predicted results and real results are different, such as profit of agent, because there is information asymmetry between principal and agent, and the principal's prediction is not completely correct in the real situation, which always plays an important role in the process of schedule risk management. But if there is no prediction, the whole project would be completely out of control; the effect of schedule risk management will be worse than it is now.

4.2.3. Performance Comparison of Algorithms. Examples 2, 3, and 4 are solved by GA, SA, and the SAGA designed in this paper to give a further test of the performance of SAGA. The convergence process of three algorithms in examples 2, 3, and 4 is shown in Figures 14–16, respectively. In these three examples, SAGA performs better than the other two algorithms in terms of convergent speed and the best fitness value. Therefore, SAGA is the best choice for the problem in examples 2, 3, and 4.

In addition, for each example, run the 3 algorithms 20 times to collect some statistics information, including maximum fitness (Max), minimum fitness (Min), average fitness (Aver), and fitness variance (Var), which is processed using the normalization method. The statistics information is shown in Table 10, where Na represents the number of agents, B represents risk management capital, A represents algorithms, ΔA represents the average fitness increasing ratio of SAGA relative to other two algorithms, and ΔV represents the difference of fitness variance between SAGA and two other algorithms.

The convergent degree of algorithm is shown by the average fitness, and the reliability of algorithm is shown by the fitness variance. The conclusion that the SAGA designed in this paper shows a certain advantages in each example can be seen from the table. With the increase of the problem scale, the convergence and reliability of SAGA are also compared with GA and SA in Figures 17 and 18, respectively. It can be seen that SAGA performs better than GA and SA on convergence and reliability, especially for the large scale problem.

Besides convergence and reliability, the running time of the algorithm is also an important index to measure the performance of the algorithm. Running time is greatly influenced by the experimental environment. The experimental environment in this experiment: software conditions: Windows 7, hardware conditions: DELL Optiplex 9020 +i7 Core, and development tools: Eclipse. Figure 19 shows the running time of the three algorithms for the four examples. From the Figure 19, it can be seen that when the problem scale is relatively small, the running time of the three algorithms is almost equal. When the scale of the problem is larger, for examples 3 and 4, the running time of SAGA is significantly longer than that of GA and SA, because SAGA takes more time to operate its complex searching process. It also can be seen that the running time of the three algorithms is always in the same order of magnitude for each case. Therefore, the running time of SAGA is acceptable.

5. Conclusion

This paper focuses on the schedule risk of IT outsourcing project, the DDM theory and principal-agent theory are applied to build a two-level principal-agent schedule risk management model for IT outsourcing project, and the SAGA is also designed to solve the resulting optimization problem. The simulation results illustrate that the model

Table 7: Parameters setting of SAGA for examples 2, 3, and 4.

Example	*NP*	*NG*	*Pc*	*Pm*	*Ti*	*Ni*
2	40	80	0.9	0.1	150	10
3	50	100	0.9	0.1	150	20
4	50	120	0.9	0.01	200	10

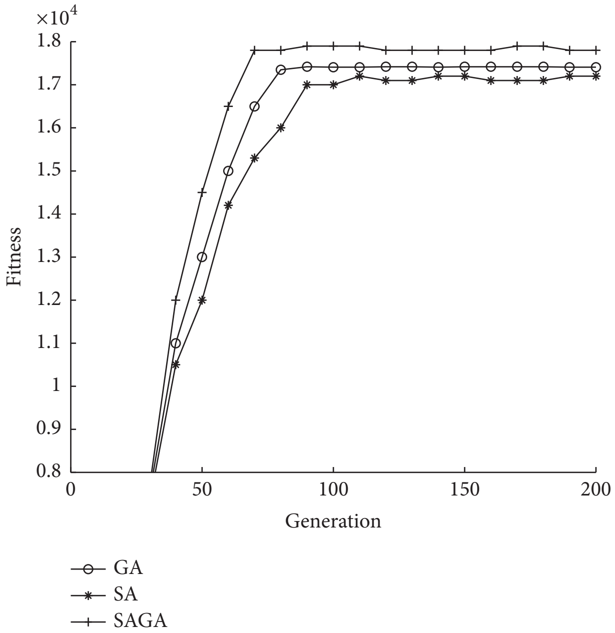


Figure 14: The convergence process of the three algorithms in example 2.

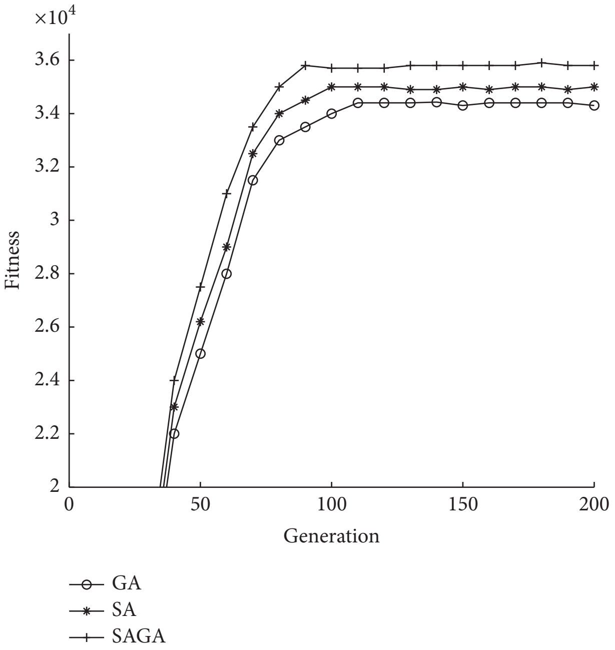


Figure 15: The convergence process of the three algorithms in example 3.

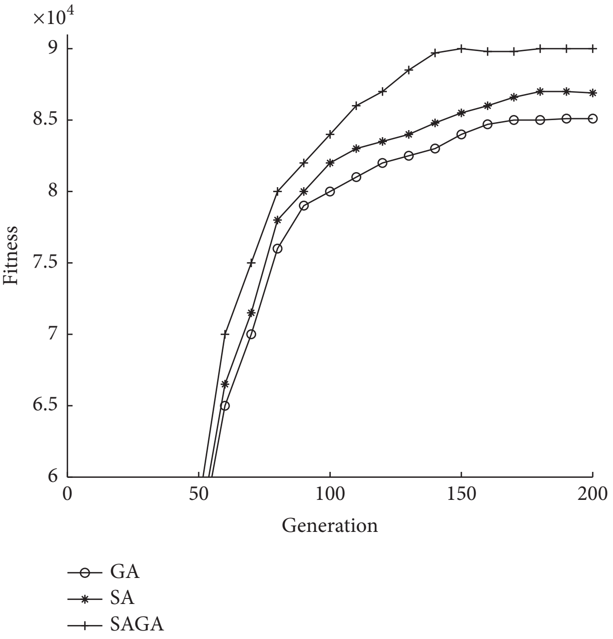


Figure 16: The convergence process of the three algorithms in example 4.

Table 8: The simulation results of example 2.

Index	Predicted results	Real results
Allocated capital	696 625 693 679 686	696 625 693 679 686
Measure combination	4 5 2 10 10 8 10	1 2 6 10 6 9 9
	4 5 2 9 6 2 10	9 3 7 3 9 3 3
	3 8 10 1 10 9 8	7 3 7 7 9 7 3
	9 1 10 1 1 10 10	7 3 7 7 9 3 7
	2 1 10 9 8 9 10	3 3 7 7 9 7 3
Duration	365.1 393.7 369.2 388.5 364.7	377.9 404.5 386.5 384.1 393.7
Finish the project before planned duration?	Y YYYY	Y N Y YY
Saved risk loss	7990.4 7988.4 7986.8 7987.5 7984.9	8000 7637.9 8000 8000 8000
Save total risk loss?	Y YYYY	Y N Y YY
Profit of agent	2494.6 2376.5 2482.9 2488.6 2505.2	2494.6 2150.6 2411.7 2409.7 2411.0
Profit of principal	17864.6	17762.6

Table 9: The simulation results of example 3.

Index	Predicted results	Real results
Allocated capital	682 579 579 646 609	682 579 579 646 609
	579 689 689 617 646	579 689 689 617 646
Measure combination	9 1 9 2 10 10 10	3 1 6 3 10 8 7
	10 2 1 9 1 7 9	4 3 10 4 1 5 9
	1 2 6 7 9 3 8	1 3 10 4 9 5 4
	5 10 2 5 10 2 10	1 3 10 4 9 5 9
	2 6 2 2 9 9 9	1 3 10 4 4 5 9
	1 3 6 1 8 9 9	1 3 10 4 4 5 9
	8 4 1 10 7 10 10	1 3 10 4 9 5 9
	1 10 10 1 10 8 9	1 3 10 10 9 5 10
	1 3 7 1 10 10 8	1 1 10 4 9 5 10
	8 2 1 8 10 9 7	1 3 10 4 10 5 10
Duration	364.0 394.9 396.6 382.5 388.3	391.4 400.1 400.1 384.7 397.1
	392.4 364.8 369.3 386.5 377.7	397.1 384.7 368.5 386.4 379.7
Finish the project before planned duration?	Y YYYY	Y N N Y Y
	Y YYYY	Y YYYY
Saved risk loss	7986.9 7988.9 7983.4 7989.5 7985.6	8000 7998.5 7925.1 8000 8000
	7988.7 7989.0 7988.0 7984.2 7986.2	8000 8000 8000 8000 8000
Save total risk loss?	Y YYYY	Y N N Y Y
	Y YYYY	Y YYYY
Profit of agent	2497.9 2355.7 2333.5 2413.4 2389.2	2451.1 2327.9 2279.9 2424.7 2366.7
	2367.6 2490.7 2502.1 2415.9 2424.9	2336.7 2467.7 2498.7 2418.4 2441.3
Profit of principal	35750.6	35707.7

Table 10: The comparison of algorithms' performance.

Example	Na	B	A	Max	Min	Aver	ΔA (%)	Var	ΔV
1	1	700	GA	2350.2	2300.8	2328.6	3.7	0.562	−0.39
1	1	700	SA	2380.7	2315.6	2344.0	2.9	0.264	−0.09
1	1	700	SAGA	2465.5	2400.2	2413.6	0	0.174	0
2	5	3500	GA	17726.1	17001.3	17419.6	2.1	0.657	−0.51
2	5	3500	SA	17466.0	17180.7	17353.8	2.5	0.196	−0.05
2	5	3500	SAGA	17911.0	17637.0	17787.3	0	0.147	0
3	10	7000	GA	35012.0	33567.9	34347.1	4.0	0.869	−0.85
3	10	7000	SA	35230.5	34616.1	35048.2	2.0	0.109	−0.09
3	10	7000	SAGA	35828.3	35546.4	35713.9	0	0.022	0
4	25	17500	GA	85000.4	84250.2	84706.0	6.1	0.399	−0.31
4	25	17500	SA	87526.4	86815.3	87236.5	2.7	0.509	−0.42
4	25	17500	SAGA	90006.3	89639.1	89860.5	0	0.092	0

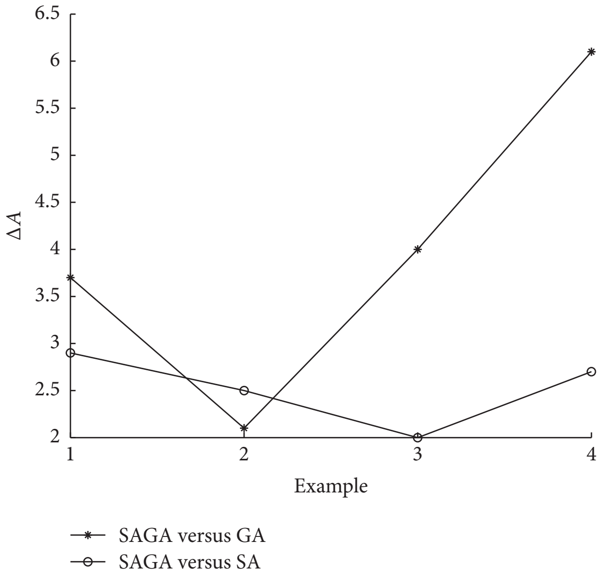


Figure 17: The convergence advantage change graph of SAGA.

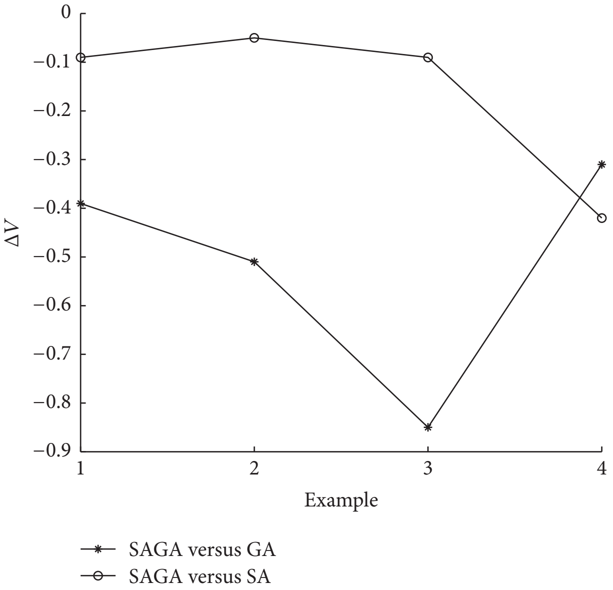


Figure 18: The stability advantage change graph of SAGA.

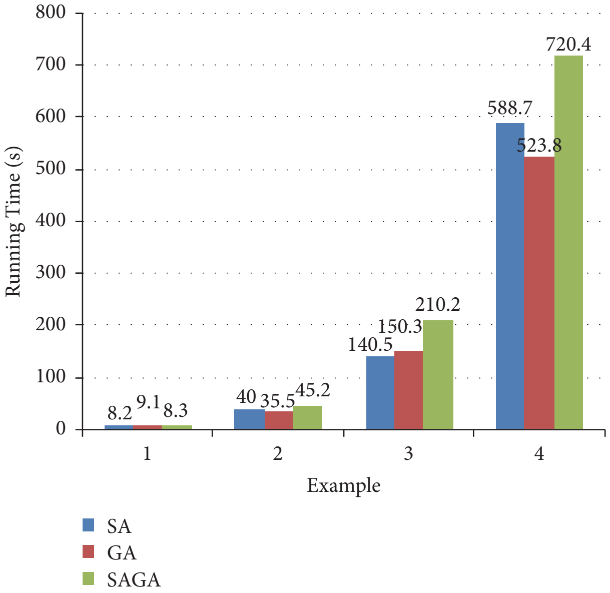


Figure 19: The running time comparison of the three algorithms.

can greatly reduce the risk loss and the duration of IT outsourcing project, which achieves the goal of managing the schedule risk effectively. In addition, the proposed SAGA is a greatly improved method, which can effectively solve the described problem comparing with the other two algorithms. Consequently, the above model and algorithm can provide important decision support for managing the schedule risk of IT outsourcing project.

Conflicts of Interest

The authors declare that the funding in Acknowledgments did not lead to any conflicts of interest regarding the publication of this manuscript.

Acknowledgments

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant no. 71325002; Major International Joint Research Project of NSFC under Grant no. 71620107003; the National Natural Science Foundation of China under Grant no. 71401027; Natural Science Foundation of Hebei Province under Grant no. G2016501086; Key Project of Science and Technology Research of Higher Education in Hebei Province under Grant no. ZD2016202.

References

[1] M. C. Lacity, S. A. Khan, and L. P. Willcocks, “A review of the IT outsourcing literature: Insights for practice,” *Journal of Strategic Information Systems*, vol. 18, no. 3, pp. 130–146, 2009.

[2] Z. P. Fan, W. Suo, and B. Feng, “Identifying risk factors of IT outsourcing using interdependent information: an extended DEMATEL method,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 3832–3840, 2012.

[3] B. Bahli and S. Rivard, “Validating measures of information technology outsourcing risk factors,” *Omega*, vol. 33, no. 2, pp. 175–187, 2005.

[4] K. Huang, “Research on risk early-warning of information technology outsoucing project based on neural network,” *Applied Mechanics and Materials*, vol. 411-414, pp. 2400–2405, 2013.

[5] Gartner, *Gartner says worldwide IT spending on pace to grow 2.1 percent in 2014*, Press release, Stamford, CT, USA, 2014.

[6] R. Kliem, “Managing the risks of offshore It development projects,” *Information Systems Management*, vol. 21, no. 3, pp. 22–27, 2004.

[7] L. M. Abdullah and J. M. Verner, “Analysis and application of an outsourcing risk framework,” *Journal of Systems and Software*, vol. 85, no. 8, pp. 1930–1952, 2012.

[8] M. Alexandrova, “Risk Factors in IT Outsourcing Partnerships: Vendors’ Perspective,” *Global Business Review*, vol. 16, no. 5, pp. 747–759, 2015.

[9] C. Samantra, S. Datta, and S. S. Mahapatra, “Risk assessment in IT outsourcing using fuzzy decision-making approach: An Indian perspective,” *Expert Systems with Applications*, vol. 41, no. 8, pp. 4010–4022, 2014.

[10] H. Samadi, S. Nazari-Shirkouhi, and A. Keramati, “Identifying and analyzing risks and responses for risk management in information technology outsourcing projects under fuzzy environment,” *International Journal of Information Technology and Decision Making*, vol. 13, no. 6, pp. 1283–1323, 2014.

[11] L. Qin, H. Wu, N. Zhang, and X. Li, “Risk identification and conduction model for financial institution IT outsourcing in China,” *Information Technology and Management*, vol. 13, no. 4, pp. 429–443, 2012.

[12] X. Huang, T. Zhao, and S. Kudratova, “Uncertain mean-variance and mean-semivariance models for optimal project selection and scheduling,” *Knowledge-Based Systems*, vol. 93, pp. 1–11, 2016.

[13] M. Earl, “The risks of outsourcing IT [J],” *Sloan Management Review*, vol. 37, no. 3, pp. 26–32, 1996.

[14] B. W. Boehm, “Software Risk Management: Principles and Practices,” *IEEE Software*, vol. 8, no. 1, pp. 32–41, 1991.

[15] L. Wang, C. Fang, C.-D. Mu, and M. Liu, “A pareto-archived estimation-of-distribution algorithm for multiobjective resource-constrained project scheduling problem,” *IEEE Transactions on Engineering Management*, vol. 60, no. 3, pp. 617–626, 2013.

[16] B. W. Boehm, “Software risk management,” *IEEE Software*, vol. 14, no. 3, pp. 1–19, 1989.

[17] B. Aubert, S. Dussault, M. Patry, and S. Rivard, “Managing the risk of IT outsourcing,” in *Proceedings of the HICSS 32 - 32nd Annual Hawaii International Conference on System Sciences*, p. 10, Maui, HI, USA.

[18] B. A. Aubert, S. Dussault et al., “Assessing IT outsourcing risk,” in *Proceedings of the 31nd Hawaii International Conference on System Sciences*, Maui, HI, USA, 1998.

[19] B. A. Aubert, S. Dussault, M. Patry et al., “outsourcing risk management at British Petroleum,” in *Proceedings of the 34nd Hawaii International Conference on System Science*, Maui, HI, USA, 2001.

[20] B. Bahli and S. Rivard, “The information technology outsourcing risk: a transaction cost and agency theory-based perspective,” *Journal of Information Technology*, vol. 18, no. 3, pp. 211–221, 2003.

[21] K. Osei-Bryson and O. K. Ngwenyama, “Managing risks in information systems outsourcing: an approach to analyzing outsourcing risks and structuring incentive contracts,” *European Journal of Operational Research*, vol. 174, no. 1, pp. 245–264, 2006.

[22] C. Sanfa, C. Kai, and Z. Bin, “Producer Services Outsourcing Risk Control Based on Outsourcing Contract Design: Industrial Engineering Perspective,” *Systems Engineering Procedia*, vol. 2, pp. 308–315, 2011.

[23] S. Xianli, H. Min, and W. Xingwei, “Distributed decision-making risk management of virtual enterprises with principal agent characteristics,” *Journal of System Engineering*, vol. 25, no. 3, 2010.

[24] S. Christoph, *Distributed Decision Making*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[25] S. Christoph, “Distributed decision making-a unified approach,” *European Journal of Operational Research*, vol. 150, no. 2, pp. 237–252, 2003.

[26] F. Sun, S. Duan, and F. Lu, “A two-level principal-agent model for schedule risk control of it outsourcing project based on genetic algorithm,” in *Proceedings of the 23rd International Conference on Industrial Engineering and Engineering Management*, pp. 17-18, 2016.

[27] C. Fang and F. Marle, “A simulation-based risk network model for decision support in project risk management,” *Decision Support Systems*, vol. 52, no. 3, pp. 635–644, 2012.

[28] C. Jerson, M. Annelies et al., “A multivariate approach for top-down project control using earned value management,” *Decision Support Systems*, vol. 79, pp. 65–76, 2015.

[29] M. Compare, F. Martini, and E. Zio, “Genetic algorithms for condition-based maintenance optimization under uncertainty,” *European Journal of Operational Research*, vol. 244, no. 2, pp. 611–623, 2015.

[30] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, “Global and local real-coded genetic algorithms based on parent-centric crossover operators,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.

[31] Y. Shuai, S. Bradley et al., “A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms,” *European Journal of Operational Research*, vol. 228, no. 1, pp. 72–82, 2013.

[32] K. Deep and M. Thakur, “A new crossover operator for real coded genetic algorithms,” *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.

[33] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Cambridge, UK, 1992.

[34] H. Ahonen, A. G. de Alvarenga, and A. R. Amaral, “Simulated annealing and tabu search approaches for the Corridor Allocation Problem,” *European Journal of Operational Research*, vol. 232, no. 1, pp. 221–233, 2014.

[35] D. Zhang, Y. Liu, R. M’Hallah, and S. C. H. Leung, “A simulated annealing with a new neighborhood structure based algorithm

for high school timetabling problems," *European Journal of Operational Research*, vol. 203, no. 3, pp. 550–558, 2010.

[36] E. Rodriguez-Tello, J.-K. Hao, and J. Torres-Jimenez, "An improved simulated annealing algorithm for bandwidth minimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1319–1335, 2008.

[37] O. Briant, D. Naddef, and G. Mounié, "Greedy approach and multi-criteria simulated annealing for the car sequencing problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 993–1003, 2008.

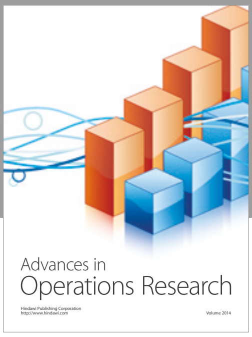
[38] Z. Dai and X. Zheng, "Design of close-loop supply chain network under uncertainty using hybrid genetic algorithm: a fuzzy and chance-constrained programming model," *Computers and Industrial Engineering*, vol. 88, pp. 444–457, 2015.

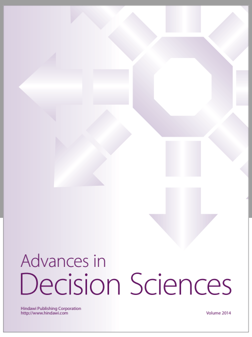
[39] M. A. Rahman and G. K. Venayagamoorthy, "A hybrid method for power system state estimation using Cellular Computational Network," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 140–151, 2017.

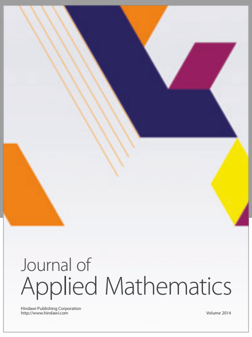
[40] N. Pérot and N. Bousquet, "Functional Weibull-based models of steel fracture toughness for structural risk analysis: estimation and selection," *Reliability Engineering & System Safety*, vol. 165, pp. 355–367, 2017.

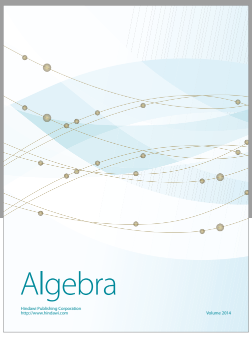
[41] N. Wattanapongskorn and D. W. Coit, "Fault-tolerant embedded system design and optimization considering reliability estimation uncertainty," *Reliability Engineering and System Safety*, vol. 92, no. 4, pp. 395–407, 2007.

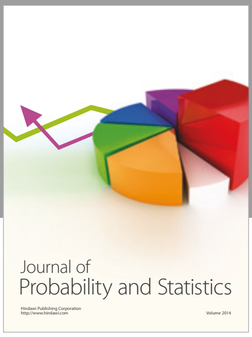
[42] S. Chatterjee and S. Bandopadhyay, "Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10943–10951, 2012.


Advances in
Operations Research

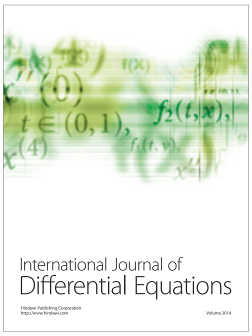

Advances in
Decision Sciences

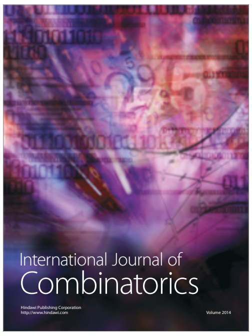

Journal of
Applied Mathematics

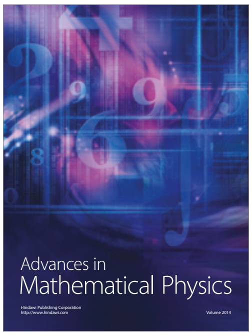

Algebra

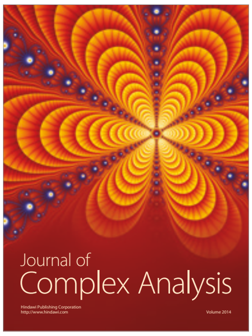

Journal of
Probability and Statistics

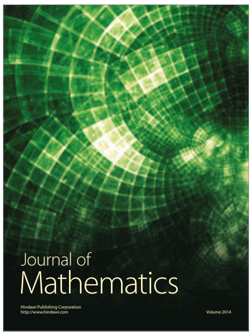
The Scientific
World Journal

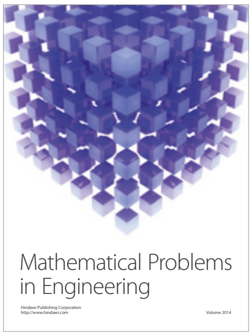

International Journal of
Differential Equations

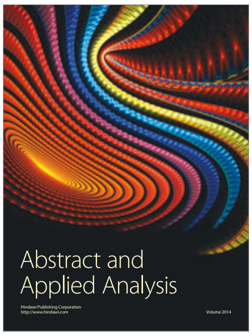

International Journal of
Combinatorics

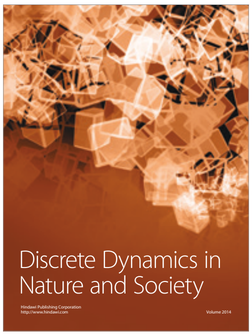

Advances in
Mathematical Physics

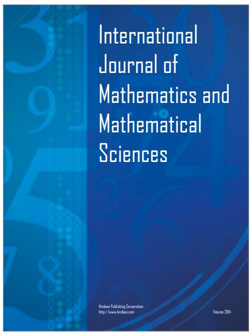

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

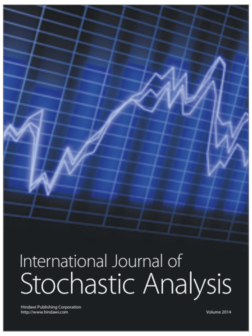

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization