Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 170 (2005) 503–514
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Models and algorithm for stochastic shortest path problem

Xiaoyu Ji

Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China

Abstract

In this paper, we study the shortest path problem with stochastic arc length. According to different decision criteria, we originally propose the concepts of expected shortest path, α-shortest path and the most shortest path, and present three new types of models: expected value model, chance-constrained programming and dependent-chance programming. In order to solve these models, a hybrid intelligent algorithm integrating stochastic simulation and genetic algorithm is developed and some numerical examples are given to illustrate its effectiveness.

Keywords: Shortest path problem; Genetic algorithm; Hybrid intelligent algorithm; Stochastic simulation

1. Introduction

The shortest path problem, finding the path with minimum distance, time or cost from a source to a destination, is one of the most fundamental problems in network theory. It arises in a wide variety of scientific and engineering problem

E-mail address: xyji@orsc.edu.cn

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.12.015

settings, both as stand-alone models and as subproblems in more complex problem settings.

The classical shortest path problem with certain arc lengths has been studied intensively. And many efficient algorithms have been developed by Bellman [2], Dijkstra [4], Dreyfus [5], and these algorithms are referred to as the standard shortest path algorithms.

However, due to failure, maintenance or other reasons, different kinds of uncertainties are frequently encountered in practice, and must be taken into account. For example, the lengths of the arcs are assumed to represent transportation time or cost rather than the geographical distances, as time or cost fluctuate with traffic or weather conditions, payload and so on, it is not practical to consider each arc as a deterministic value. In these cases, probability theory has been used to attack randomness, and many researchers have done lots of work on stochastic shortest path problem. Frank [6], Mirchandani [15] and Sigal et al. [18] studied the probability distribution of the shortest path length in which arc lengths are random variables. Loui [14], Murthy and Sarkar [16] considered the different types of cost functions to study the variations of the shortest path problem in stochastic networks. Hall [9] and Fu [7] studied the expected shortest paths in dynamic and stochastic networks.

In this paper, we propose three concepts of stochastic shortest path: expected shortest path, α-shortest path and the most shortest path and formulate three models for the stochastic shortest path according to difference decision criteria. In order to solve these models, a hybrid intelligent algorithm combined genetic algorithm and stochastic simulation is given for obtaining shortest paths, and this algorithm has been proved to be useful for solving practical problems by number of examples.

This paper is organized as follows: in Section 2, the problem is described in detail and in Section 3–5, three new types of concepts and stochastic models are presented, respectively. In Section 6, a hybrid intelligent algorithm integrating stochastic simulation and GA is designed. Then in Section 7, some numerical examples are given to reveal the effectiveness of the hybrid intelligent algorithm. Finally, some conclusions are drawn.

2. Stochastic shortest path problem

In order to formulate the mathematical models for stochastic shortest path problem, we give a directed acyclic network $\mathscr{G} = (\mathscr{V}, \mathscr{A})$, consisting of a finite set of nodes $\mathscr{V} = \{1, 2, \ldots, n\}$ and a set of arcs $\mathscr{A}$, in which the lengths of the arcs are assumed to be stochastic. Each arc is denoted by an ordered pair (i,j), where $(i, j) \in \mathscr{A}$. It is supposed that there is only one directed arc (i,j) from i to j. Moreover, the nodes in an acyclic directed network $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ can be renumbered so that $i < j$ for all $(i,j) \in \mathscr{A}$ [1][17]. Now, we use the following path representation:

$$\boldsymbol{x} = \{x_{ij} | (i,j) \in \mathscr{A}\},$$

where $x_{ij} = 1$ means that the arc (i,j) is in the path, $x_{ij} = 0$ means that the arc (i,j) is not in the path.

It has been proved that $\boldsymbol{x} = \{x_{ij} | (i,j) \in \mathscr{A}\}$ is a path from nodes 1 to n in a directed acyclic graph if and only if

$$\sum_{(i,j)\in\mathscr{A}} x_{ij} - \sum_{(j,i)\in\mathscr{A}} x_{ji} = \begin{cases} 1, & i = 1, \\ 0, & 2 \leqslant i \leqslant n-1, \\ -1, & i = n, \end{cases}$$

$$x_{ij} = 0 \text{ or } 1 \text{ for any} (i,j) \in \mathscr{A}.$$

Suppose that ξ_{ij} are the lengths of arcs $(i,j) \in \mathscr{A}$, respectively. We write $\boldsymbol{\xi} = \{\xi_{ij} | (i,j) \in \mathscr{A}\}$, then the length of path $\boldsymbol{x}$ is

$$T(\boldsymbol{x}, \xi) = \sum_{(i,j)\in\mathscr{A}} \xi_{ij} x_{ij}.$$

3. Expected shortest path model

In practice, the decision-makers want to find the path with the minimum expected time or cost to help them make decision. Firstly, we give the following definition:

Definition 1. A path $\boldsymbol{x}$ is called the expected shortest path if

$$E[T(\boldsymbol{x}, \boldsymbol{\xi})] \leqslant E[T(\boldsymbol{x}', \boldsymbol{\xi})]$$

for all paths $\boldsymbol{x}'$ from nodes 1 to n in $\mathscr{G}$, where $E[T(\boldsymbol{x}, \boldsymbol{\xi})]$ is called the expected shortest path length.

In order to find the expected shortest path, we give the following expected value model of stochastic shortest path problem shown below:

$$\begin{cases} \min \ E\left[\sum_{(i,j)\in\mathscr{A}} \xi_{ij} x_{ij}\right] \\ \text{subject to :} \\ \quad \sum_{(1,j)\in\mathscr{A}} x_{1j} - \sum_{(j,1)\in\mathscr{A}} x_{j1} = 1, \\ \quad \sum_{(i,j)\in\mathscr{A}} x_{ij} - \sum_{(j,i)\in\mathscr{A}} x_{ji} = 0, \ \ 2 \leqslant i \leqslant n-1, \\ \quad \sum_{(n,j)\in\mathscr{A}} x_{nj} - \sum_{(j,n)\in\mathscr{A}} x_{jn} = -1, \\ \quad x_{ij} \in \{0,1\}, \ \ \forall (i,j) \in \mathscr{A}. \end{cases} \tag{1}$$

4. The most shortest path model

Sometimes, the decision-makers want to maximize the chance functions of some events (i.e., the probabilities of satisfying the events). In the shortest path problem, there is generally a predetermined cost or time T_0, which decision-makers can accept. Therefore, we should consider the path with greatest chance to be shorter than the given T_0. We give the following definition:

Definition 2. A path $\boldsymbol{x}$ is called the most shortest path if

$$\Pr\{T(\boldsymbol{x},\boldsymbol{\xi}) \leqslant T_0\} \geqslant \Pr\{T(\boldsymbol{x}',\boldsymbol{\xi}) \leqslant T_0\}$$

for all paths $\boldsymbol{x}'$ from nodes 1 to n in $\mathscr{G}$, where T_0 is a predetermined cost or time.

In order to model this type of stochastic decision system, Liu [12] provided a type of stochastic programming, called dependent-chance programming (DCP), in which the underlying philosophy is based on selecting the decision with maximal chance to meet the event. So we have the following DCP model for the most shortest path:

$$\begin{cases} \max\ \Pr\left\{\sum_{(i,j)\in\mathscr{A}}\xi_{ij}x_{ij} \leqslant T_0\right\} \\ \text{subject to}: \\ \quad \sum_{(1,j)\in\mathscr{A}}x_{1j} - \sum_{(j,1)\in\mathscr{A}}x_{j1} = 1, \\ \quad \sum_{(i,j)\in\mathscr{A}}x_{ij} - \sum_{(j,i)\in\mathscr{A}}x_{ji} = 0, \ \ 2 \leqslant i \leqslant n-1, \\ \quad \sum_{(n,j)\in\mathscr{A}}x_{nj} - \sum_{(j,n)\in\mathscr{A}}x_{jn} = -1, \\ \quad x_{ij} \in \{0,1\}, \ \ \forall (i,j) \in \mathscr{A}. \end{cases} \tag{2}$$

5. α-Shortest path model

Sometimes, the decision-makers are interested in the path which satisfies some chance constraints with at least some given confidence level α, so we have the following definition:

Definition 3. A path $\boldsymbol{x}$ is called the α-shortest path from nodes 1 to n if

$$\min\{\bar{T} \mid \Pr\{T(\boldsymbol{x},\boldsymbol{\xi}) \leqslant \bar{T}\} \geqslant \alpha\} \leqslant \min\{\bar{T} \mid \Pr\{T(\boldsymbol{x}',\boldsymbol{\xi}) \leqslant \bar{T}\} \geqslant \alpha\}$$

for any path $\boldsymbol{x}'$ from nodes 1 to n, where $T(\boldsymbol{x},\boldsymbol{\xi})$ is called α-length of path, α is predetermined confidence level.

In order to find the α-shortest path, we adopt the chance-constrained programming (CCP) which is developed by Charnes and Cooper [3] and Liu [13]. CCP offers a powerful means of modelling stochastic decision systems with assumption that the stochastic constraints will hold at least α times, where α is referred to as the confidence level provided as an appropriate safety margin by the decision-maker. We give the model for the α-shortest path as follows:

$$\begin{cases} \min \bar{T} \\ \text{subject to :} \\ \quad \Pr\left\{\sum_{(i,j)\in\mathscr{A}} \xi_{ij} x_{ij} \leqslant \bar{T}\right\} \geqslant \alpha, \\ \quad \sum_{(1,j)\in\mathscr{A}} x_{1j} - \sum_{(j,1)\in\mathscr{A}} x_{j1} = 1, \\ \quad \sum_{(i,j)\in\mathscr{A}} x_{ij} - \sum_{(j,i)\in\mathscr{A}} x_{ji} = 0, \quad 2 \leqslant i \leqslant n-1, \\ \quad \sum_{(n,j)\in\mathscr{A}} x_{nj} - \sum_{(j,n)\in\mathscr{A}} x_{jn} = -1, \\ \quad x_{ij} \in \{0,1\}, \quad \forall (i,j) \in \mathscr{A}. \end{cases} \tag{3}$$

6. Hybrid intelligent algorithm

Generally speaking, stochastic programming models are difficult to solve by traditional methods. In order to solve above models, we develop a hybrid intelligent algorithm integrating stochastic simulation and genetic algorithm.

6.1. Computing uncertain functions

Uncertain functions mean their parameters are stochastic. Due to the complexity, we design some stochastic simulation to estimate uncertain functions in our models. For convenience, $\boldsymbol{\xi}$ is presented in another way as $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_m)$, where m is the number of the arc. At first, we calculate the following function:

$$U_1 : \boldsymbol{x} \rightarrow E[T(\boldsymbol{x}, \boldsymbol{\xi})].$$

Here we design a stochastic simulation as follows:

Step 1. Set $U_1(\boldsymbol{x}) = 0$.
Step 2. Generate $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_m)$ from the distribution functions.
Step 3. $U_1(\boldsymbol{x}) \leftarrow U_1(\boldsymbol{x}) + T(\boldsymbol{x}, \omega)$.
Step 4. Repeat the second and third steps N times.
Step 5. $U_1(\boldsymbol{x}) \leftarrow U_1(\boldsymbol{x})/N$.

Secondly, let us simulate the following uncertain function:

$$U_2 : \boldsymbol{x} \rightarrow \Pr\{T(\boldsymbol{x}, \boldsymbol{\xi}) \leqslant T_0\}.$$

We make N trials by producing ξ_n, $n = 1, 2, \ldots, N$. Let N' denote the number of occasions on which $T(\boldsymbol{x}, \xi) \leqslant T_0$ (i.e., the number of ξ satisfying the inequalities). Let us define

$$h(\xi) = \begin{cases} 1, & \text{if } T(\boldsymbol{x}, \xi) \leqslant T_0, \\ 0, & \text{otherwise,} \end{cases}$$

then we have $E[h(\xi_n)] = U_2$, and $N' = \sum_{n=1}^{N} h(\xi_n)$. It follows from the strong law of large numbers that:

$$\frac{N'}{N} = \frac{\sum_{n=1}^{N} h(\xi_n)}{N}$$

converges a.s. to U_2. Thus the probability U_2 can be estimated by N'/N provided that N is sufficiently large. So this kind of uncertain function can be estimated by the following procedure:

Step 1. Set $N' = 0$.
Step 2. Generate $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_m)$ from the distribution function of ξ.
Step 3. If $T(\boldsymbol{x}, \omega) \leqslant T_0$, then $N' + +$.
Step 4. Repeat the second and third steps N times.
Step 5. $U_2 = N'/N$.

The last type of uncertain function in our optimization problem is

$$U_3 : \boldsymbol{x} \to \min\{\bar{T} | \Pr\{T(\boldsymbol{x}, \xi) \leqslant \bar{T}\} \geqslant \alpha\}.$$

In order to find the minimum $\bar{T}$ satisfying $\Pr\{T(\boldsymbol{x}, \xi) \leqslant \bar{T}\} \geqslant \alpha$, we define

$$h(\xi) = \begin{cases} 1, & \text{if } T(\boldsymbol{x}, \xi) \leqslant \bar{T}, \\ 0, & \text{otherwise.} \end{cases}$$

By the strong law of large numbers, we obtain

$$\frac{\sum_{n=1}^{N} h(\xi_n)}{N} \to \alpha, \quad \text{a. s.}$$

as $N \to \infty$. Note that the sum $\sum_{n=1}^{N} h(\xi_n)$ is just the number of ξ satisfying $T(\boldsymbol{x}, \xi) \leqslant \bar{T}$. Thus the value $\bar{T}$ can be taken as the N'th largest element in the sequence $\{T(\boldsymbol{x}, \xi_1), T(\boldsymbol{x}, \xi_2), \ldots, T(\boldsymbol{x}, \xi_N)\}$, where N' is the integer part of αN. This type of uncertain function can be estimated by the following procedure:

Step 1. Generate $\omega_1, \omega_2, \ldots, \omega_N$ from the distribution functions, where N is a sufficiently large number.
Step 2. Set $T_i = T(\boldsymbol{x}, \xi_i)$ for $i = 1, 2, \ldots, N$.
Step 3. Set N' as the integer part of $(1 - \alpha)N$.
Step 4. Return the N'th largest element in $\{T_1, T_2, \ldots, T_N\}$.

6.2. Genetic algorithm

Ever since the genetic algorithm was introduced by Holland [11] to tackle combinatorial problems, it has emerged as one of the most efficient stochastic solution search procedures for solving various network design problem [8].

In order to solve above models of stochastic shortest path problem, we employ genetic algorithm to find the paths. The representation structure, initialization and genetic operators are as follows.

6.2.1. Genetic representation

Now, we use an integer vector $P = (v_1, v_2, \ldots, v_k)$ as a chromosome to represent a path of $\mathscr{G}$ from nodes 1 and n. Because different paths include different notes and arcs, the dimension of chromosome is not fixed. If $(v_1, v_2, \ldots, v_k)$ represents a path from nodes 1 to n, then we have $(1, v_1) \in \mathscr{A}$, $(v_1, v_2) \in \mathscr{A}, \ldots, (v_{k-1}, v_k) \in \mathscr{A}$ and $(v_k, n) \in \mathscr{A}$. So we have the following definition:

$$x_{ij} = \begin{cases} 1, & \text{if } i = 1, j = v_1, \\ 1, & \text{if there exists } l \text{ such that } i = v_l, j = v_{l+1}, \\ 1, & \text{if } i = v_k, j = n, \\ 0, & \text{otherwise,} \end{cases}$$

for all $(i, j) \in \mathscr{A}$. It is also easy to verify that $\{x_{ij} | (i, j) \in \mathscr{A}\}$ obtained by this way is a path from nodes 1 to n. Conversely, let $\{x_{ij} | (i, j) \in \mathscr{A}\}$ be a path from nodes 1 to n. We may obtain a chromosome by the following procedure.

6.2.2. Chromosome initialization

In order to initialize a feasible chromosome, we adopt the following heuristic procedure:

Step 1. Set $l = 0$ and $v_0 = 1$.
Step 2. Randomly choose an index m such that $(v_l, m) \in \mathscr{A}$.
Step 3. $l \leftarrow l + 1$ and $v_l = m$.
Step 4. Repeat the second and third steps until $v_l = n$.
Step 5. Obtain a chromosome $(v_1, v_2, \ldots, v_{l-1})$.

6.2.3. Genetic operators

Genetic operators mimic the process of heredity of genes to generate new offspring at each generation and play a very important role in genetic algorithm

[10]. In our algorithm, the crossover operator, mutation operator and selection are as follows:

Chromosome crossover. Let $P_1 = (v_1, v_2, \ldots, v_k)$ and $P_2 = (v'_1, v'_2, \ldots, v'_{k'})$ be two chromosomes. We will do crossover operation on them as follows: if there are common nodes between them, then we randomly choose one, say $v_i = v'_{i'}$. We produce the following two chromosomes:

$$(v_1, v_2, \ldots, v_i, v'_{i'+1}, \ldots, v'_{k'}), \quad (v'_1, v'_2, \ldots, v'_{i'}, v_{i+1}, \ldots, v_k),$$

which are also feasible chromosomes representing paths from nodes 1 to n. If there is no common node, then do nothing.

Chromosome mutation. Let $P = (v_1, v_2, \ldots, v_k)$ be a chromosomes, we can mutate it by the following way. Generate an integer from $\{1, 2, \ldots, k\}$ randomly, denoted by i. Then we make a path $(v'_{i+1}, \ldots, v'_{k'})$ from v_i to n by a similar process of chromosome initialization, and produce a new chromosome $(v_1, v_2, \ldots, v_i, v'_{i+1}, \ldots, v'_{k'})$.

Selection. The roulette wheel selection is adopted in our algorithm. We select a single chromosome each time for a new population until *pop_size* copies are finally obtained.

6.3. Hybrid intelligent algorithm

We integrate stochastic simulations and genetic algorithm to produce a hybrid intelligent algorithm. The algorithm can be described as follows:

Step 1. Initialize *pop_size* chromosomes P_k, $k = 1, 2, \ldots, pop\_size$ at random.

Step 2. Calculate the objective values for all chromosomes by stochastic simulations, respectively.

Step 3. Compute the fitness of each chromosome. The rank-based evaluation function is defined as

$$\mathrm{Eval}(P_i) = a(1-a)^{i-1}, \quad i = 1, 2, \ldots, pop\_size,$$

where the chromosomes are assumed to have been rearranged from good to bad according to their objective values and $a \in (0, 1)$ is a parameter in the genetic system.

Step 4. Select the chromosomes for a new population.

Step 5. Update the chromosomes P_k, $k = 1, 2, \ldots, pop\_size$ by crossover operation and mutation operation mentioned in the last subsection.

Step 6. Repeat the second to fifth steps for a given number of cycles.

Step 7. Report the best chromosome $P^* = (v_1, v_2, \ldots, v_k)$.

7. Numerical examples

In order to illustrate the hybrid intelligent algorithm, we consider a transportation network with 23 nodes and 40 directed arcs shown in Fig. 1. Lengths of arcs are represented as random variables, and their distribution functions are shown in Table 1, in which $\mathcal{N}(\mu, \sigma^2)$ means normally distribution, $\mathcal{U}(a, b)$ means uniformly distribution, EXP(β) means exponentially distribution and $\mathcal{T}(a, b, m)$ means triangularly distribution.

The hybrid intelligent algorithm will be run on a personal computer with the following parameters: *pop_size* is 30, the number of generations is 800, the number of stochastic simulation cycles is 5000, the probability of crossover is 0.2, the probability of mutation is 0.2, and the parameter *a* in the rank-based evaluation function is 0.05. To solve the model (1), a run of the hybrid intelligent algorithm shows that the expect shortest path is $1 \to 5 \to 11 \to 17 \to 21 \to 23$, and the length of this path is 46.34.

With the same parameters of genetic algorithm, we run the hybrid intelligent algorithm for model (2), and we can get the most shortest paths as follows:

$T_0 = 70$: $1 \to 3 \to 8 \to 12 \to 15 \to 18 \to 21 \to 23$, probability = 0.015;
$T_0 = 75$: $1 \to 3 \to 8 \to 12 \to 15 \to 18 \to 21 \to 23$, probability = 0.154;
$T_0 = 80$: $1 \to 4 \to 7 \to 10 \to 17 \to 20 \to 23$, probability = 0.566;
$T_0 = 85$: $1 \to 4 \to 7 \to 10 \to 17 \to 20 \to 23$, probability = 0.658;
$T_0 = 90$: $1 \to 4 \to 7 \to 10 \to 17 \to 20 \to 23$, probability = 0.756;
$T_0 = 95$: $1 \to 4 \to 7 \to 10 \to 17 \to 20 \to 23$, probability = 0.823;
$T_0 = 100$: $1 \to 4 \to 7 \to 10 \to 17 \to 20 \to 23$, probability = 0.956.

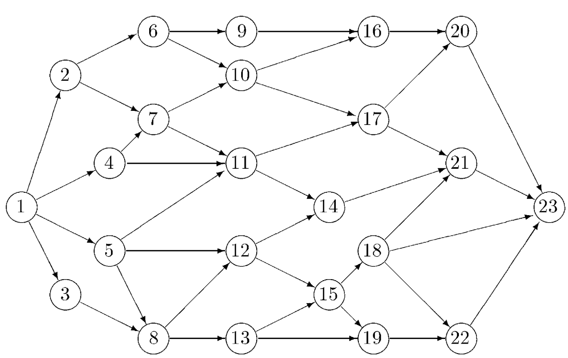

Fig. 1. A transportation network.

Table 1
Lengths of arcs

Arc	Distribution function	Arc	Distribution function	Arc	Distribution function
(1,2)	$N(12,1)$	(1,4)	$T(8,12,10)$	(1,5)	$N(7,1)$
(1,3)	$N(9,1)$	(2,6)	$T(5,15,7)$	(2,7)	$U(6,11)$
(3,8)	$U(10,16)$	(4,7)	EXP(20)	(4,11)	$T(6,13,10)$
(5,11)	$U(7,13)$	(5,12)	EXP(13)	(5,8)	$T(6,10,9)$
(6,9)	$T(6,10,8)$	(6,10)	$T(10,14,11)$	(7,10)	$U(9,12)$
(7,11)	$U(6,8)$	(8,12)	$U(5,9)$	(8,13)	$N(5,1)$
(9,16)	EXP(7)	(10,16)	$U(12,16)$	(10,17)	$T(15,19,17)$
(11,17)	EXP(9)	(11,14)	$N(9,1)$	(12,14)	$T(10,15,13)$
(12,15)	$N(12,2)$	(13,15)	$T(10,14,12)$	(13,19)	$T(17,19,18)$
(14,21)	$N(11,1)$	(15,18)	$T(8,11,9)$	(15,19)	$N(7,1)$
(16,20)	$T(9,12,10)$	(17,20)	$T(7,12,11)$	(17,21)	$U(6,8)$
(18,21)	$N(15,2)$	(18,23)	$T(5,9,7)$	(18,22)	EXP(5)
(19,22)	$U(15,17)$	(20,23)	$T(13,15,14)$	(21,23)	$T(12,15,13)$
(22,23)	$U(4,6)$				

Similarly, we run the hybrid intelligent algorithm for solving the α-shortest path model (3) with different confidence levels α, and the α-shortest paths are

0.9-shortest path: $1 \to 5 \to 8 \to 13 \to 15 \to 18 \to 23$, 0.9-length = 51.26;
0.8-shortest path: $1 \to 5 \to 8 \to 13 \to 15 \to 18 \to 23$, 0.8-length = 50.48;
0.7-shortest path: $1 \to 5 \to 11 \to 17 \to 21 \to 23$, 0.7-length = 47.25;
0.6-shortest path: $1 \to 5 \to 11 \to 17 \to 21 \to 23$, 0.6-length = 45.02.

In order to illustrate the robust of this hybrid intelligent algorithm, as an example, we solve the model (3) with the confidence level 0.85. We select different parameters of genetic algorithm, and use relative error as the index, i.e., (actual value − minimum value)/minimum value $\times$ 100%, which are shown in Table 2. It follows from Table 2 that the relative error does not exceed 3.0%

Table 2
Comparison solutions of the 0.85-shortest path

	α	*Pop_size*	P_c	P_m	*Gen*	Simulation times	Path lengths	Error (%)
1	0.05	20	0.3	0.1	500	5000	51.08	0.42
2	0.05	20	0.3	0.2	500	5000	51.27	1.35
3	0.05	20	0.1	0.3	800	4000	50.27	1.45
4	0.05	30	0.2	0.2	800	5000	50.50	2.47
5	0.05	20	0.2	0.1	500	4000	51.19	3.00
6	0.02	30	0.3	0.2	500	4000	50.43	0.66
7	0.02	30	0.1	0.3	500	4000	50.42	0.22
8	0.02	30	0.3	0.2	500	5000	50.49	0.00
9	0.02	30	0.2	0.3	500	5000	50.30	1.17
10	0.02	30	0.1	0.2	500	4000	50.49	2.78

when different parameters are selected, which implies that the hybrid intelligent algorithm is robust to the parameters and effective to find the 0.85-shortest path.

8. Conclusion

In this paper, we have contributed to the shortest path problem in the following aspects. First, three types of stochastic programming models were presented for the first time: expected shortest path model, the most shortest path model and α-shortest path model. Second, to solve these stochastic models, a hybrid intelligent algorithm based on genetic algorithm and stochastic simulation is also developed. Then some numerical examples were given to show the performance of the hybrid intelligent algorithm. The results of these examples show that this algorithm is robust to the parameters of genetic algorithm, and it is possible to solve problems with large number of nodes and arcs.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 60174049), and Specialized Research Fund for the Doctoral Program of Higher Education (No. 20020003009).

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice-Hall, Englewood CliRs, NJ, 1993.
[2] E. Bellman, On a routing problem, Q. Appl. Math. 16 (1958) 87–90.
[3] A. Charnes, W.W. Cooper, Chance-constrained programming, Manag. Sci. 6 (1959) 73–79.
[4] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.
[5] S. Dreyfus, An appraisal of some shortest path algorithms, Oper. Res. 17 (1969) 395–412.
[6] H. Frank, Shortest paths in probability graphs, Oper. Res. 17 (1969) 583–599.
[7] L. Fu, L.R. Rilett, Expected shortest paths in dynamic and stochastic traffic networks, Transp. Res. 32 (7) (1998) 499–516.
[8] M. Gen, R. Cheng, Genetic Algorithms and Engineer Optimization, John Wiley and Sons Inc., New York, 2000.
[9] R. Hall, The fastest path through a network with random time-dependent travel time, Transp. Sci. 20 (3) (1986) 182–188.
[10] C. Hsinghua, G. Premkumar, C. Chu, Genetic algorithms for communications network design—an empirical study of the factors that influence performance, IEEE Trans. Evol. Comput. 5 (3) (2001) 236–249.
[11] J. Holland, Adaptatin in Natural and Artificial System, University of Michigan Press, Ann Arbor, MI, 1975.

[12] B. Liu, Dependent-chance programming: a class of stochastic programming, Comp. Math. Appl. 34 (1997) 89–104.
[13] B. Liu, Theory and Practice of Uncertain Programming, Physica-Verlag, Heidelberg, 2002.
[14] P. Loui, Optimal paths in graphs with stochastic or multidimensional weights, Commun. ACM 26 (1983) 670–676.
[15] B.P. Mirchandani, Shortest distance and reliability of probabilistic networks, Comp. Oper. Res. 3 (1976) 347–676.
[16] I. Murthy, S. Sarkar, A relaxation-based pruning technique for a class of stochastic shortest path problems, Transp. Sci. 30 (3) (1996) 220–236.
[17] K.G. Murty, Network Programming, Prentice-Hall, Englewood CliRs, NJ, 1992.
[18] C.E. Sigal, A.A.B. Pritsker, J.J. Solberg, The stochastic shortest route problem, Oper. Res. 28 (5) (1980) 1122–1129.