Reliability based assignment in stochastic-flow freight network

Wangtu Xu a,*, Shiwei He b, Rui Song c, Juan Li d

a School of Traffic and Transportation, Beijing Jiaotong University, 3 # 801, Xue Yuan Student Apartment, Beijing 100044, China
b Department of Transportation Management, School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
c Department of Traffic Engineering, School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
d School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

A R T I C L E   I N F O

Keywords: System reliability Stochastic-flow network Transportation assignment Chance-constraint programming Hybrid intelligent algorithm

A B S T R A C T

Based on the reliability of transportation time, a transportation assignment model of stochastic-flow freight network is designed in this paper. This transportation assignment model is built by mean of stochastic chance-constraint programming and solved with a hybrid intelligent algorithm (HIA) which integrates genetic algorithm (GA), stochastic simulation (SS) and neural network (NN). GA is employed to report the optimal solution as well as the optimal objective function values of the proposed model. SS is used to simulate the value of uncertain system reliability function. The uncertain function approximated via NN is embedded into GA to check the feasibility and to compute the fitness of the chromosomes. These conclusions have been drawn after a test of numerical case using the proposed formulations. System reliability, total system cost and flow on each path would finally reach at their own convergence points. Increase of the system reliability causes increase of the total time cost. The system reliability and the total time cost converge at a possible Nash Equilibrium point.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Transportation assignment (TA) is the process in which the transport demand (either the passenger or the freight) between the origin node (O) and the destination node (D) in the network are loaded on the paths connecting O and D. For TA, a key work is to enumerate the connecting paths between O and D [1]. The traditional path enumerations were according to a series of classical graph theory algorithms such as Dijkstra’s Shortest Path Problem (SP), Ford algorithm, and Floyd algorithm, etc. [1]. The K shortest paths (KSP) and the feasible paths extended from the SP were also frequently employed in TA [2]. Recently, the quickest paths problem (QPP) which was extended from the KSP and aimed at enumerating the paths which have the minimal transportation times between an OD pair, also attached many researchers’ attentions [3,4]. Chen and Huang [5], Hung and Chen [6], Lin et al. [7] discussed the constrained quickest path problem. They all proposed their own methods to enumerate the quickest paths considering different constraints. Rosen et al. extended the existing researches and they studied the K quickest paths problem (KQPP) [8]. Chen finished the similar work [9]. Lee and Papadopoulou formulated the architecture of all-pairs quickest path problem (A-PQPP), and proposed the concept of the Kth all-pairs quickest path with the enumeration algorithm [10].

Almost all of the researches presented above have a common foundation. That is, they are all based on a deterministic network whose capacities of the arcs are fixed values. Unfortunately, the real-life networks (e.g., the communication network, the power supply network, the water supply network or the transportation network) always have the stochastic...
arc capacities. Network whose arc capacities are stochastic is defined as the stochastic-flow network (SFN) [11–13]. In the context of SFN, a key factor to evaluate the network performance is system reliability (SR), which is represented by the probability that the system capacity is not less than the demand, and has been evaluated in terms of various graph theories since last decade [7,11,12,14–17].

Besides SR evaluations, optimization problems involved in SFN also caused many researches' interests. Optimization issues focusing on SFN could be divided into three types, i.e., reliability optimization, strategy optimization and robust optimization of SFN. Reliability optimization of SFN is to design a network at a high level reliability with satisfactory capacity, performance or level of service. Chen and Zhou proposed a model and an algorithm for reliability optimization in generalized stochastic-flow networks. Their algorithm used the relationship of the k-weak-link sets and sets of failure events to the parameters of the generalized stochastic-flow networks [18]. Considering the optimal reliability, Hsieh and Lin studied the multi-resource allocation in a stochastic-flow network. And, an algorithm incorporating the principle of minimal path vectors is used to solve the resource allocation problem [19]. The strategy optimization problems of SFN are the most popular issues in SFN studies. Such as, optimal flow strategy on stochastic transportation network by Doulliez and Jamoulle [20]; the optimal routing strategy of a SFN by Lin [21]; stochastic fluid model by Sun et al. [22], etc. [23]. Robust optimization of SFN is based on the reliability optimization and the strategy optimization. It always objectives at designing a SFN system with the robust reliability, strategy solutions. Relatively few studies have explored the robust optimization of SFN. Beyer and Sendhoff [24] presented the comprehensive approaches of robust optimizations for SFN. Barbarosogcaronlu and Arda [25] studied the robust optimization of framework of transportation planning in disaster SFN situations.

In this paper, we analyse the SR of a stochastic-flow freight network (SFFN) whose arc capacities are stochastic and follow assumed distributions. We defined the system reliability of SFFN as the probability that SFFN could successfully transport a given amount of freights between O and D. Based on the SR of SFFN, we study the freight transportation assignment algorithm.

Our problem integrates the reliability optimization and the strategy optimization on SFN. Our objective is to find out the strategy that how much flow should be assigned on each path connecting the OD pairs with the minimal time elapse and the maximal successful transportation probability (that is, the system reliability). We analyze the possible flow assignment strategies, and determine the optimal strategy via comparing the system reliability and the transportation time cost of these strategies. That is, our optimization aims at the optimal flow assignment strategy with the highest level system reliability. We optimize the strategy and the reliability at the same time.

We used a stochastic change-programming model [26,27] to formulate the freight flow assignment model. For the change-programming model, Charnes and Cooper [26] have certified it as a NP-hard problem. In order to determine an optimal flow assignment strategy via change-programming, we need to design some heuristic algorithm. Since genetic algorithm (GA) is an effective algorithm for finding the optimal solution of change-programming [27,28], we also embed the stochastic simulation (SS), which can simulate the value of uncertain system reliability function, as well as the neural network (NN), which can be used to approximate the uncertain function of system reliability, into GA to design a hybrid intelligent algorithm (HIA). The HIA could be performed in the following way. We generate a large number of training data of the uncertain function value of system reliability via SS. Using these training data in NN, we approximate an uncertain function of system reliability. Then the approximated uncertain function could be embedded into GA to check the feasibility and compute the fitness of chromosomes. And, after several times of iterations in GA, the optimal flow assignment solution could be found out.

The rest of this paper has been organized as follows. In the next section we describe the basic assumptions and the symbol definitions. We introduce the concept of the system reliability of SFFN in Section 3. The freight transportation assignment algorithm for SFFN is formulated in Section 4 and its solution algorithm is in Section 5. In Section 6 the proposed assignment algorithm is tested by a numerical case and its solution efficiency is analyzed. In the last section we draw our conclusions, as well as the future research directions.

2. Basic assumptions and symbol definitions

2.1. Basic assumptions

(1) Only one type of freight is transported in SFFN. That is, the SFFN in this paper is a mono-commodity network.

(2) Capacity of each arc of SFFN is stochastic, and follows a given probability distribution.

(3) Capacities of any two arcs are independent on each other.

(4) Flows in all paths between OD satisfy the feasible flow principle.

2.2. Symbol definitions

(G(V, A)) a stochastic-flow freight network, where V represents the set of nodes, v ∈ V is a node; A is the set of the arcs and a ∈ A denotes arc a of the network

L system reliability of the freight transportation in a given time constraint in SFFN

N set of OD pairs in SFFN
3. System reliability of the stochastic-flow freight network

3.1. Path capacity and transportation time

For the $k$th path of the $n$th OD pair $k_n = \{a_{k_n}^1, a_{k_n}^2, \ldots, a_{k_n}^j, \ldots, a_{k_n}^{A_{k_n}}\}$, where $a_{k_n}^j$ denotes the $j$th arc included in path $k_n$, $j = 1, 2, \ldots, A_{k_n}$, we define its capacity as

$$c_{k_n} = \min_j \{\zeta_{a_{k_n}^j}\}.$$ (1)

If $d$ units’ freight is transported by path $k_n$, we define the transportation time, $R(d, \xi, k_n)$ as

$$R(d, \xi, k_n) = \sum_{a \in A} \delta_{a_{k_n}^j} d_{a_{k_n}^j}^{\xi} + \alpha \left( \frac{d}{\zeta_{a_{k_n}^j}} \right)^\beta,$$ (2)

where $R(d, \xi, k_n)$ represents the time of transporting $d$ units’ freight by path $k_n$ under the stochastic arc capacity vector $\xi$. $\alpha$ and $\beta$ are user-defining parameters [1].

Thus, the transportation time of the $q_n$ units’ freight between OD pair $n$ could be determined as the maximal path transportation time

$$Q(n, q_n, \xi) = \max_{k_n \in K_n} R(q_n, \xi, k_n).$$ (3)

3.2. System reliability of the stochastic-flow freight network

For a capacity vector $\xi$, event $Q(n, q_n, \xi) \leq T$ denotes SFFN could successfully transport $q_n$ units’ freight between OD pair $n$ in time constraint $T$. We could introduce the concept of the time reliability of transporting $q_n$ units’ freight between OD pair $n$ as

$$L_n = \text{Pr}\{\xi | Q(n, q_n, \xi) \leq T\},$$ (4)

where Pr{} is a probability measure [27].

Based on Formula (4), we define the system reliability of SFFN on transporting $Q$ freights between all OD pairs as

$$L = \prod_{n \in N} L_n = \prod_{n \in N} \text{Pr}\{\xi | Q(n, q_n, \xi) \leq T\},$$ (5)

where $Q$ is the freight demand matrix, $Q = \{q_1, q_2, \ldots, q_n\}$.

4. Demand assignment algorithm for SFFN

Due to the stochastic arc capacity vector, the traditional transportation assignment algorithm such as user equilibrium method (UE), system optimal method (SO), etc. [1] which have been employed frequently in the deterministic freight network, could not be used in SFFN anymore. Consequently, we apply the uncertainty theory [27] to the model formulation of transportation assignment in SFFN.

In the context of the stochastic arcs capacities, the decision-maker would aim at transporting the freight with the minimal time under a requested SR confidence level. On the other hand, the decision-maker would pursue the minimal total time cost in the entire SFFN.
Firstly, make sure that SR of SFFN should be higher than a given confidence level \( \gamma \).

\[
\Pr\{\xi|Q(N, \xi) \leq T\} \geq \gamma, \quad \forall n \in N. \tag{6}
\]

Formula (6) could be written in goal programming representation as follows

\[
\Pr\{\xi|Q(N, \xi) \leq T\} + d^-_n = \gamma, \quad \forall n \in N, \tag{7}
\]

where \( d^-_n \) and \( d^+_n \) denote the minus and positive deviations for the first goal set by the decision-maker. \( \gamma \) is the confidence level of the SR.

Secondly, pursue the minimal total time cost in the entire network, that is

\[
\sum_{a \in A} x_at_a(x_a) + d^-_n - d^+_n = 0, \tag{8}
\]

where \( t_a(x_a) = t^0_a[1 + \alpha (\frac{x_a}{c_a})^\beta] \) is the time cost of the arc on which the flow is \( x_a \). \( t^0_a \) is the free-flow time cost of arc \( a \) (details of free-flow time cost could be seen in [1]). \( \alpha \) and \( \beta \) are user-defining parameters, in this paper they are set equal to 0.15 and 2.0 respectively. \( d^-_n \) and \( d^+_n \) are the minus and positive deviations for the second goal.

According to the basic assumptions in Section 2.1 and OD assignment condition [1], we should obey the flow conservation condition

\[
\sum_{k \in K_n} f^n_k = q_n, \quad \forall n \in N. \tag{9}
\]

The flows on the arcs relate the flows on the paths between all OD pairs as

\[
x_a = \sum_{n \in N} \sum_{k \in K_n} \delta^n_{kn} f^n_k, \quad \forall a \in A. \tag{10}
\]

The path flows should not be minuses, such that

\[
f^n_k \geq 0, \quad \forall k \in K_n, \quad \forall n \in N. \tag{11}
\]

Based on the double level objective function values and the constrained conditions, we formulate transportation assignment model for SFFN under the stochastic arc capacity situation as

\[
\begin{aligned}
\text{lex min} & \{p^1d^-_n, p^2d^+_n\} \\
\text{s.t.} & \Pr\{\xi|Q(N, \xi) \leq T\} + d^-_n - d^+_n = \gamma, \\
& \sum_{a \in A} x_at_a(x_a) + d^-_n - d^+_n = 0, \\
& \sum_{k \in K_n} f^n_k = q_n, \quad \forall n \in N, \\
& x_a = \sum_{n \in N} \sum_{k \in K_n} \delta^n_{kn} f^n_k, \quad \forall a \in A, \\
& f^n_k \geq 0, \quad \forall k \in K_n, \quad \forall n \in N, \\
& d^-_n, d^+_n \geq 0.
\end{aligned} \tag{12}
\]

In Formula (12), \( \text{lexmin} \) represents lexicographically minimizing the objective vector. The objective function includes two levels of goal values. The first goal is to minimize \( d^-_n \) to make sure the SR of SFFN should be larger than \( \gamma \). The second goal is to minimize \( d^+_n \) to make the total time cost as low as possible, where \( p^i, i = 1, 2 \), are the positive preemptive priority factors which express the relative importance of the two goals.

5. Solution algorithm

Formula (12) is formulated as a chance-constraint programming which is one type of stochastic programming [27]. Generally, we could not convert the chance-constraint programming into a deterministic form. There is no deterministic algorithm to solve this type of formulation. Liu [27] designed a hybrid intelligent algorithm (HIA) for it. This algorithm integrates stochastic simulation (SS), neural network (NN), genetic algorithm (GA) and has been adopted to solve many types of stochastic programming [27–29]. The framework of this HIA is shown in Fig. 1.

Based on the HIA in Fig. 1, we design a solution algorithm for Programming (12).

Step. 0 Design the uncertain function as \( U(X) : X \rightarrow \Pr\{\xi|Q(N, \xi) \leq T\} \). Code the chromosome as \( X = \{x[1]|1], x[1]|2], \ldots, x[n]|k], \ldots, x[n]|2N\} \), where \( x[n]|k] \) denotes the freight flow on the \( k \)th path of the \( n \)th OD pair.

Step. 1 Generate training input–output data for uncertain function \( U(X) : X \rightarrow \Pr\{\xi|Q(N, \xi) \leq T\} \) by SS of [27], where we have \( \sum_{n=1}^N \Omega_n \) inputs and one output. Each flow on a path is an input and the value of uncertain function \( U(X) \) is the only output.
Step 0. Design the uncertain function, and set the chromosome code, the chromosome feasibility standard and the genetic operation for the GA.

Step 1. Generate training input-output data for uncertain functions like \( U : x \rightarrow \Pr \{ \bullet \} \) by stochastic simulation.

Step 2. Train a neural network to approximate the uncertain function according to the generated training input-output data.

Step 3. Initialize pop_size chromosomes whose feasibility may be checked by the given constrained conditions.

Step 4. Update the chromosomes by crossover and mutation operations in which the feasibility of offspring need not be checked. Because the chromosomes generated by Formula (13) necessarily satisfy the constraints of Formula (9) to Formula (11). We apply the crossover operation and mutation operation of Literature [27] for this HIA.

Step 5. Calculate the objective values for all chromosomes by the trained neural network.

Step 6. Compute the fitness of each chromosome according to the objective values.

Step 7. Select the chromosomes by spinning the roulette wheel.

Step 8. Repeat the fourth to seventh steps for a given number of cycles.

Step 9. Report the best chromosome as the optimal solution.

Fig. 1. Framework of hybrid intelligent algorithm [27].

Step 2. Train a NN to approximate the uncertain function according to the generated training input–output data.

Step 3. Initialize pop_size chromosomes whose feasibility should be checked by the constrained conditions of Formula (9) to Formula (11). We generate the chromosomes by

\[
\begin{align*}
    x[n][1] &= \text{myu}(0, q_n) \\
    x[n][2] &= \text{myu}(0, q_n - x[n][1]) \\
    &\vdots \\
    x[n][k] &= \text{myu}(0, q_n - x[n][1] - x[n][2] - \cdots x[n][k-1]) \\
    &\vdots \\
    x[n][\Omega_n - 1] &= \text{myu}(0, q_n - x[n][1] - x[n][2] - \cdots x[n][\Omega_n - 2]) \\
    x[n][\Omega_n] &= q_n - x[n][1] - x[n][2] - \cdots x[n][\Omega_n - 1]
\end{align*}
\]  

(13)

where \( \text{myu}(x_{\min}, x_{\max}) \) represents a random number which follows a Uniform distribution in range \([x_{\min}, x_{\max}]\).

Step 4. Update the chromosomes by crossover and mutation operations in which the feasibility of offspring should be checked by the given constrained conditions. We apply the crossover operation and mutation operation of Literature [27] for this HIA.

Step 5. Calculate the objective values for all chromosomes. The first objective value \( d_{i_1} \) could be calculated based on the uncertain function value \( U(X) \) which is the output of the trained NN. For chromosome \( X, d_{i_1} = \{y - U(X)\} \geq 0 \), where \( \geq \) is the maximization function. The second objective function value \( d_{i_2} \) could be calculated by

\[
d_{i_2} = \sum_{a=A}^{B} t_a(x_a) \geq 0.
\]

Step 6. Compute the fitness of each chromosome based on the objective values. The fitness of the chromosome is calculated via the evaluation function in [30] and based on the two objective values, \( d_{i_1} \) and \( d_{i_2} \). If two chromosomes have the same \( d_{i_1} \) for this double objective programming, we define a preference function to evaluate the chromosomes. We set preemptive priority factors \( \rho^1 \) and \( \rho^2 \) of two goals \( (d_{i_1} \text{ and } d_{i_2}) \) equal to 2 and 1 respectively. That is, we think the system reliability of freight transportation is twice as important as the total time cost. So we have the following orders for the chromosomes. For any two chromosomes, if the higher priority objectives \( (d_{i_1}) \) are equal, in the lower priority level \( (d_{i_2}) \), the one with minimal objective value is better. If two different chromosomes have the same objective values at two levels, then we are indifferent between them.

Step 7. Select the chromosomes by spinning the roulette wheel (see [27]).

Step 8. Repeat the fourth to seventh steps for a given number of cycles.

Step 9. Report the best chromosome as the optimal solution.

6. Numerical experiment

A SFFN in Fig. 2 is employed to test the efficiency of the proposed model. There are one OD pair \((s-t)\), six paths, six points and nine arcs. Capacity of each arc is stochastic. The concerned variables of the arcs are shown in Table 1.
to each other. Take this two paths total time cost. However, in this paper, we could not demonstrate that this convergence point is a Nash Equilibrium point.

That is,

$$X = \{a_1, a_6, a_9\}$$

Path 2 = \{a_1, a_3, a_4, a_8\}

Path 3 = \{a_1, a_3, a_4, a_7, a_9\} Path 4 = \{a_2, a_4, a_8\}

Path 5 = \{a_2, a_4, a_7, a_9\} Path 6 = \{a_2, a_4, a_5, a_6, a_9\}

Fig. 2. SFFN of the numerical example.

Table 1

<table>
<thead>
<tr>
<th>$\xi_i$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_i$</td>
<td>N(30, 2)</td>
<td>N(50, 5)</td>
<td>N(10, 1)</td>
<td>N(20, 1.5)</td>
<td>N(30, 0.5)</td>
<td>N(40, 1)</td>
<td>N(30, 6)</td>
<td>N(50, 10)</td>
<td>N(50, 1)</td>
</tr>
</tbody>
</table>

In Table 1, $N(\mu, \sigma^2)$ denotes the arc capacity following a Normal distribution whose expected value is $\mu$ and variance is $\sigma^2$.

In this SFFN, there is only one OD pair. So we design the chromosome as $X = \{x_1, x_2, x_3, x_4, x_5\}$, where $x_i$, $i = 1, 2, \ldots, 6$, represents the flow on path $i$ between the single OD pair $s-t$. The confidence level of the system reliability is set equal to $95\%$. That is, $\gamma = 95\%$. The required transportation time, $T$, is set equal to 80 units. Let the freight demand between the OD pair equal to 100 units, i.e., $q_1 = 100$. We can write the uncertain function $U(X) = \Pr\left[\sum_{i=1}^{9} q_i Q(1, 100, \xi_i) \leq 80\right]$. It denotes the probability of SFFN in Fig. 2 could successfully transport 100 units’ freights in 80 units’ time constraint between OD pair $s-t$.

We employ stochastic simulation to generate 3000 input–output data for the uncertain function $U(X) = \Pr\left[\sum_{i=1}^{9} q_i Q(1, 100, \xi_i) \leq 80\right]$. Then we train a NN (6 input neurons, 15 hidden neurons, 1 output neuron) to approximate the uncertain function. Finally, we combine the trained NN and GA to produce a HIA.

A run of the HIA (5000 cycles in stochastic simulation, 3000 training data in NN, 300 population size, 5000 generations, 0.1 crossover probabilities and 0.3 mutation probabilities in GA) shows the optimal solution and the objective function values

$$X^* = \{1.5963, 3.3768, 7.6942, 42.5488, 3.5982, 41.1857\};$$

$$(d_1^*, d_2^*) = (5.6967\%, 4529.698).$$

According to the first objective function value, $d_1^*$, we can calculate the optimal system reliability, $L = 95\% - 5.6967\% = 89.304\%$. We know that this SFFN could transport 100 units’ freights between $s$ and $t$ in 80 time units with a maximal successful probability of 89.304%, and needs a total time cost of 4529.698 units.

Iteration processes of $d_1^*, d_2^*$, $L$, flows of the six paths $X$, are shown in Fig. 3–6.

As we can see in Figs. 3 and 4, both the two objective values, $d_1^*$ and $d_2^*$ have their own convergence points after 2650 genetic generations. These results certify that Programming (12) has an optimal solution. In Fig. 3, $d_1^*$ decreases with the proceeding of GA iteration. We could say that the updating of the chromosomes results in the decrease of minus division between the system reliability and its confidence level (95%). In Fig. 4, $d_2^*$ increases with the proceeding of GA iteration, which demonstrates that the increase of the system reliability would cause the increase of the total time cost. According to Fig. 3, $L = 95\% - d_1^*$, the iteration process of the system reliability $L$ is shown in Fig. 5.

We could say that the optimal solution of Programming (12) is a similar Nash Equilibrium solution. Increase of system reliability would result in the increase of total time cost. The optimal solution compromises the system reliability and the total time cost. However, in this paper, we could not demonstrate that this convergence point is a Nash Equilibrium point.

In Fig. 6, the iteration processes of the six paths show that flows on the fourth path (Path 4) and the sixth path (Path 6) evolve dramatically with the processing of GA generations. And, flows on the other four paths (Path 1, Path 2, Path 3, Path 5) are relatively stable. Flow of each path converges at its own point after 2650 generations. Before 2650 generations, we obtain a best chromosome of $X = \{3.3998, 3.8738, 10.2838, 65.5929, 0.2907, 16.1591\}$ and a objective function value vector of $(d_1^*, d_2^*) = (6.4027\%, 4088.7487)$. At 2648 generation, flow on Path 4 decreases and flow on Path 6 increases sharply to close to each other. Take this two paths (Path 4 = \{a_2, a_6, a_9\}, Path 6 = \{a_2, a_6, a_5, a_9\}) into account, Path 4 is the minimal cost
path, and Path 6 shares two arcs ($a_2$ and $a_6$) with P4. Capacity variance of arc $a_8$ is larger than that of arc $a_5$, $a_6$ and $a_9$. The flow evolutions on Path 4 and Path 6 tell us that the freight should be transported by the minimal cost paths first. Then the flow on the minimal cost path would increase and its capacity might be reduced so that the congestion might happen. If still use this path to transport the freight, the system reliability would be very small. So we would choose a path with the smallest stochastic capacity variance to transport an amount of freight. That is, we would rather choose a path with larger

---

**Fig. 3.** Iteration process of the first objective function value $d_1^-$.  

**Fig. 4.** Iteration process of the second objective function value $d_2^+$.  

---
transportation time reliability and pay higher time cost than using a path with smaller time reliability with lower time cost after the minimal cost path has been congested.

The evolution processes of the path flows ulteriorly demonstrate that the optimal solution might be a Nash Equilibrium solution, which could transport the freight in SFFN with the highest system reliability and reduce the total time cost to the best of the decision-maker’s ability.

We test the traditional UE assignment in this freight network and deem this network as the deterministic network whose arc capacities equal to the expected values of the arc capacities in Table 1. Keeping other parameters as the same, we finally obtain a solution of $X = (0, 0, 0, 100, 0, 0)$, which is an UE solution [1] and its system reliability is only 84.3%. This tells us that the UE assignment strategy could not embody the Nash Equilibrium characteristic in SFFN. It minimizes the total time cost while the failure risk of freight transportation is higher and it’s relatively speculative in transporting freight.

Fig. 5. Iteration process of system reliability $L(L = 95\% - d_t)$.

Fig. 6. Iteration processes of the flows on six paths.
In order to check the computation efficiency of the proposed model, we run the HIA with three types of parameters strategies. The three types of parameters strategies are to analyze the running time changes of the HIA if some parameters (we choose the number of training data for NN, iterative generations in GA, population size of chromosomes in GA) are changed. The three parameters strategies are respectively as follows. For the first strategy, 3000 training data for NN, 300 population size of the chromosomes in GA (denoted as GA-pop_size), the iterative generations in GA (denoted as GA-Generations) is changed, set from 1000 to 10,000. The second strategy changes the number of training data from 1000 to 10,000, and keeps GA-Generations and GA-pop_size as 5000 and 300 respectively (kept the same as primary). The third strategy changes GA-pop_size from 10 to 6000, and keeps GA-Generations and the number of training data as 5000 and 3000 as primary. We run 10 times HIA with the three parameter strategies repetitively. That is, keep other parameters the same as the primary, run 10 times of HIA with the changes of GA-Generations from 1000 to 10,000, the number of training data from 1000 to 10,000, GA-pop_size from 10 to 6000 respectively. On a computer with Pentium 2.8 GHz CPU, 512 MB RAM, times (calculated as the CPU time) of running thirty times (one strategy 10 times, respectively) of HIA are shown in Fig. 7.

In Fig. 7, there are three abscissas. The two at the bottom are for GA-Generations and the number of training data, and the upper one is scaled for GA-pop_size.

As it could be seen in Fig. 7, CPU times of HIAs change versus the changes of GA-Generations, GA-pop_size and the number of training data. When changing GA-Generations from 1000 to 10,000, the CPU time of running HIA increases smoothly from 313.1 s to 466.2 s. Whereas, it increases relatively rapidly from 301.0 s to 900.0 s when changing the GA-pop_size from 10 to 6000. And that, the increase of CPU time is very sharp when running the HIAs via changing the number of training data from 1000 to 10000. It increases from 197.8 s to 264.5 s. The profiles of these three curves in Fig. 7 tell us that the running time of the HIA lies largely on the number of training data of NN, and secondly on the population size of chromosomes in GA. And, the iterative generations in GA does not have many effects on the running time of HIA. Consequently, choosing an appropriate training data number for NN is important. The small number of training data might result in large errors when approximating the uncertain function of system reliability using the NN. While, just as shown in Fig. 7, the large number of training data would cause the dramatic increase of running time in finding the optimal solution. After 100 times of the experiments of running the HIA, we find 3000 training data is most proper for the algorithm in this paper.

Fig. 7. CPU times of running the HIA using different parameters strategies.
7. Conclusions

Based on the system reliability of transportation time, we study the transportation assignment in a stochastic-flow network. We formulate a stochastic chance-constraint programming model for the transportation assignment. A hybrid intelligent algorithm embedding stochastic simulation, neural network and genetic algorithm is employed to solve this programming. We use a numerical case to certify the feasibility of the proposed model. The analytical results demonstrate that increases of the system reliability causes the increase of the total transportation time cost; and that system reliability and the total transportation time cost converge to a possible Nash Equilibrium point.

The future researches would focus on the following issues.

(1) Study whether the convergence point of the system reliability and the total time cost is a Nash Equilibrium point and analyse the further relationship between them.

(2) Formulate transportation assignment model for multi-commodities, multi-classes SFFN, rather than SFFN with only one type of freight.

Acknowledgement

This research has been supported in part by Chinese National Programs for High Technology Research and Development (863 Programs) under Grant 2007AA11Z208.

References