

Research Article

Assessing the Reliability of a Multistate Logistics Network under the Transportation Cost Constraint

Qi Zeng and Xiu-Zhen Xu

School of Business Administration, Henan Polytechnic University, Jiaozuo 454000, China

Correspondence should be addressed to Xiu-Zhen Xu; jluxxz@qq.com

Received 12 April 2016; Accepted 26 July 2016

Academic Editor: Ricardo López-Ruiz

Copyright © 2016 Q. Zeng and X.-Z. Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reliability and transportation cost are two important indicators to measure the performance of logistics network. As a combination of reliability and transportation cost, the performance index $R_{(d,c)}$ of logistics network is defined as the probability that at least d units of flow demand can be successfully transmitted from the source to the destination with the total transportation cost less than or equal to c . In this paper, an algorithm is developed to calculate $R_{(d,c)}$ in terms of (d, c) -minimal paths ((d, c) -MPs for short). The proposed algorithm employs a decomposition technique to divide the search space of (d, c) -MPs such that the search space of (d, c) -MPs can be dramatically reduced, and thereby (d, c) -MPs can be efficiently obtained. An example is provided to illustrate the proposed algorithm. Finally, computational experiments conducted on one benchmark network indicate that the proposed method has an advantage over the existing methods.

1. Introduction

With the swift development of E-commerce, express logistics in China is presenting a vigorous development trend. Meanwhile, most of enterprises begin to pay more attention to logistics performance for improving business efficiency. Logistics networks which provide the infrastructure for the storage and distribution of products undertake the mission of promoting the efficient and safe movement of goods over time and space and thus play an increasingly important part in sustaining the national economic and social development. It is widely accepted that an efficient, reliable, and cost-effective logistics network not only affects a firm's short-term benefits but also influences a firm's long-term development.

The performance assessment of logistics networks is a popular issue in the field of logistics and supply chain management. In a real-world environment, the performance of logistics networks is always affected by various unexpected events, such that it is subject to degradation. Hence, it is of importance from the perspective of logistics management to assess the capability of logistics networks to ensure the delivery of required quantity of goods to the right demand point. In addition, the transportation cost is also a major

concern for logistics providers. Therefore, combining capacity reliability and transportation cost for performance measure of logistics networks, in a sense, is of significance to possible improvement of logistics performance.

Theoretically, logistics network can be represented as sets of both nodes and arcs, where each node stands for a supplier, a transfer center, or a market, and each arc connecting a pair of nodes stands for transportation medium (traffic tools, traffic routes, or both) [1]. In fact, the transportation medium, which is multistate due to the nature of the traffic tools (truck, railway, cargo ship, etc.), may be in a failure state, partial failure state, or maintenance state [1, 2]. That is, the number of available traffic tools is not fixed in some sense, and thus each arc has several possible capacities. Therefore, logistics network can be modeled as a multistate flow network in which arcs are associated with multiple integer capacities, operational reliability, and unit transportation cost. And, the goods transported through such logistics network are looked upon as a flow. The reliability index $R_{(d,c)}$ of logistics network is defined as the probability that at least d units of flow demand can be successfully transmitted from the source to the destination with the total transportation cost less than or equal to c [3–9]. One of the general algorithms for computing

$R_{(d,c)}$ is using (d, c) -minimal paths $((d, c)$ -MPs) [4–9]. A (d, c) -MP, x , is a minimal state vector meeting the demand d and the cost constraint c , which means that, for any $y < x$, y does not meet the demand d or the cost constraint c [6]. If all (d, c) -MPs are known, the well-known Inclusion-Exclusion rule is available to calculate $R_{(d,c)}$ [7–9]. So, the most important work is how to efficiently determine all (d, c) -MPs.

A minimal path (MP) is a subset of arcs, such that if any arc is removed from this set, the remaining set is no longer a path [10]. Based on MPs, Lin [5] proposed a simple algorithm to search for all (d, c) -MPs. Lin's algorithm first uses the enumeration algorithm to search for (d, c) -MP candidates. Then, all (d, c) -MP candidates are verified whether they are (d, c) -MPs via the comparison method. It is time consuming to check (d, c) -MP candidates by a comparison method due to the exponentially growing number of (d, c) -MP candidates [7]. Lin [6] extended the work to the case with unreliable nodes. The method of Lin uses MPs to assign the flow to each component (arc or node). All (d, c) -MPs can be obtained by the comparison algorithm. Instead of the comparison method, Yeh [7] proposed a cycle-checking method to verify (d, c) -MP candidates and demonstrated that his method is more efficient. Without requiring MPs information, Yeh [8] proposed an algorithm to search for (d, c) -MPs by solving a simple model and proved that his algorithm is more efficient than the MPs based algorithms [5–7]. And yet, the algorithm by Yeh [8] is an exhaustive enumeration method in itself, so its searching efficiency is low. By adding some constraints to the minimal capacities of arcs, Niu and Xu [9] proposed an algorithm to improve the searching efficiency of solving (d, c) -MPs.

The above discussions indicate that the methods in [5–7] for solving (d, c) -MPs require all MPs information. However, it is a cumbersome task to find all MPs, because the MP problem is proven to be NP-hard [11]. Furthermore, the time complexity of the MPs based algorithms [5–7] is directly proportional to the number of MPs which grows exponentially with the scale of the network; thus, the computational efficiency of these algorithms in [5–7] is far from satisfactory. The methods in [8, 9] require no MPs information and thus are an improvement to the ones in [5–7]. But, one remarkable limitation of Yeh's method [8] is that a huge number of state vectors need to be enumerated, which results in the lower searching efficiency. While Niu and Xu [9] have made some efforts to advance the searching efficiency, the method by Niu and Xu is even inferior to the one by Yeh when the demand level is low, and thus its searching efficiency is also not satisfactory. Hence, the purpose of this paper is to propose a new efficient algorithm for solving all (d, c) -MPs. The major contribution of this paper is that we develop a new approach to shorten the search space of (d, c) -MP. In particular, to decrease the number of enumerated state vectors, an effective decomposition technique is employed to divide the search space of (d, c) -MPs, such that the search subspace of (d, c) -MPs can be dramatically reduced; thereby (d, c) -MPs can be efficiently obtained. A real example is provided to illustrate the proposed algorithm. Moreover, the experimental results clearly indicate that the proposed algorithm is superior to

the above-mentioned methods [5–9] in solving the (d, c) -MP problem.

The rest of this paper is organized as follows. Section 2 introduces the network model and basic results. In Section 3, the proposed algorithm is described in detail, and its time complexity is also analyzed. In Section 4, an illustrative example is provided to demonstrate the proposed algorithm. Computational experiments are conducted in Section 5 to compare the proposed algorithm with the existing methods. The final section presents the concluding remarks.

2. Preliminaries

2.1. Multistate Network Model. A multistate network is represented by $G(V, E, u, C)$ with the source node s and the destination node t , where $V = \{s, 1, 2, \dots, n, t\}$ is the set of nodes with n denoting the number of nodes except s and t , $E = \{e_1, e_2, \dots, e_m\}$ is the set of arcs with m denoting the number of arcs, $u = (u_1, u_2, \dots, u_m)$ is the largest capacity vector, and $C = (C_1, C_2, \dots, C_m)$ is the cost vector, where C_i is the unit transportation cost of e_i . The state of an arc e_i is denoted by x_i which takes integer values from 0 to u_i , where u_i denotes the largest capacity of arc e_i . A state vector $x = (x_1, x_2, \dots, x_m)$ indicates the current state of each arc. Let X denote a set of state vectors, and the smallest and largest state vectors in set X are denoted by $l^X = (l_1^X, l_2^X, \dots, l_m^X)$ and $u^X = (u_1^X, u_2^X, \dots, u_m^X)$, respectively. Let $M(x)$ denote the max-flow of the network under x and f_i denote the flows through e_i for $1 \leq i \leq m$; then the total transportation cost of the network under x is $C(x) = \sum_{i=1}^m f_i C_i$. Specifically, if x is a feasible state vector (i.e., it satisfies the flow conservation law), then $C(x) = \sum_{i=1}^m x_i C_i$. The network model satisfies the following assumptions [7–9]: (1) the state of each arc is a random variable which takes integer values from 0 to u_i according to a given distribution; (2) the states of different arcs are statistically independent; (3) all flows in the network obey the conservation law: that is, total flows into and from a node (other than the source and destination nodes) are all equal.

To facilitate the understanding, Figure 1 and Table 1 are used to illustrate several notations. Figure 1 shows $V = \{s, 1, 2, t\}$, $n = 2$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, and $m = 6$. Table 1 indicates $u = (3, 2, 1, 1, 2, 2)$ and $C = (3, 1, 1, 1, 1, 3)$. Given a state vector $x = (2, 1, 1, 0, 1, 2)$, which indicates that the current states of e_1, e_2, e_3, e_4, e_5 , and e_6 are 2, 1, 1, 0, 1, and 2, respectively, the max-flow of the network under x is $M(x) = 3$. Obviously, $x = (2, 1, 1, 0, 1, 2)$ is a feasible state vector; then the total transportation cost of the network under x is $C(x) = 2 \times 3 + 1 \times 1 + 1 \times 1 + 0 \times 1 + 1 \times 1 + 2 \times 3 = 15$. Let $X = \{(x_1, x_2, x_3, x_4, x_5, x_6) \mid x_1 = 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1, 0 \leq x_4 \leq 1, 0 \leq x_5 \leq 1, 1 \leq x_6 \leq 2\}$; then the smallest state vector in X is $l^X = (1, 0, 0, 0, 0, 1)$, and the largest state vector in X is $u^X = (1, 1, 1, 1, 1, 2)$. The max-flow of the network under l^X and u^X is $M(l^X) = 0$ and $M(u^X) = 2$.

2.2. Evaluation of $R_{(d,c)}$ in terms of (d, c) -MPs. The Inclusion-Exclusion method is introduced to show how to calculate $R_{(d,c)}$ in terms of (d, c) -MPs. Assume $y_1, y_2, \dots, y_\sigma$ are all

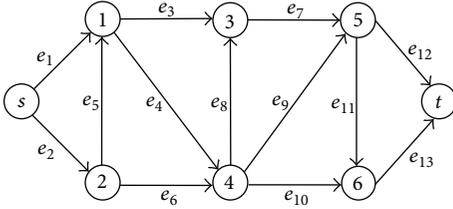


FIGURE 1

TABLE 1: The data of arcs for Figure 1.

Arc	State/capacities				State/capacity probabilities				Distribution cost
e_1	0	1	2	3	0.05	0.10	0.25	0.60	3
e_2	0	1	2	—	0.10	0.30	0.60	—	1
e_3	0	1	—	—	0.10	0.90	—	—	1
e_4	0	1	—	—	0.10	0.90	—	—	1
e_5	0	1	2	—	0.10	0.10	0.80	—	1
e_6	0	1	2	—	0.05	0.25	0.70	—	3

(d, c) -MPs, and let $B_1 = \{x \mid x \geq y_1\}$, $B_2 = \{x \mid x \geq y_2\}, \dots, B_\sigma = \{x \mid x \geq y_\sigma\}$, where $x = (x_1, x_2, \dots, x_m)$, $y_i = (y_{i1}, y_{i2}, \dots, y_{im})$, and $x \geq y_i$ means that $x_j \geq y_{ij}$ for $j = 1, 2, \dots, m$ [6]; then $R_{(d,c)}$ can be evaluated via the Inclusion-Exclusion method as follows:

$$\begin{aligned}
 R_{(d,c)} &= \Pr(B_1 \cup B_2 \cup \dots \cup B_\sigma) \\
 &= \sum_{i=1}^{\sigma} \Pr(B_i) - \sum_{j=2}^{\sigma} \sum_{i=1}^{j-1} \Pr(B_i \cap B_j) + \dots \\
 &\quad + (-1)^{\sigma-1} \Pr(B_1 \cap B_2 \cap \dots \cap B_\sigma),
 \end{aligned} \quad (1)$$

where $\Pr\{B_i\} = \Pr\{x \mid x \geq y_i\} = \prod_{k=1}^m \Pr\{x_k \geq y_{ik}\}$, $\Pr\{B_i \cap B_j\} = \Pr\{x \mid x \geq \max\{y_i, y_j\}\} = \prod_{k=1}^m \Pr\{x_k \geq \max\{y_{ik}, y_{jk}\}\}, \dots$, $\Pr(B_1 \cap B_2 \cap \dots \cap B_\sigma) = \Pr\{x \mid x \geq \max\{y_1, y_2, \dots, y_\sigma\}\} = \prod_{k=1}^m \Pr\{x_k \geq \max\{y_{1k}, y_{2k}, \dots, y_{\sigma k}\}\}$.

2.3. Fundamental Results on the (d, c) -MP Problem. A state vector $x = (x_1, x_2, \dots, x_m)$ is a (d, c) -MP if and only if (1) $M(x) = d$; (2) $M(x - 0(e_i)) < d$ for each $x_i > 0$, where $0(e_i) = (0, \dots, 0, 1, 0, \dots, 0)$; that is, state is 1 for e_i and 0 for other arcs; and (3) $C(x) = \sum_{i=1}^m x_i C_i \leq c$ [8, 9]. The following lemma which is originated from the work of Niu and Xu [9] is a basis for searching for (d, c) -MPs.

Lemma 1. A state vector x is a (d, c) -MP if and only if x satisfies the following conditions:

$$\sum_{e_i \in (s, \cdot)} x_i = \sum_{e_j \in (\cdot, t)} x_j = d, \quad (2)$$

$$\sum_{e_i \in (\cdot, v)} x_i = \sum_{e_j \in (v, \cdot)} x_j \quad \text{for every node } v \in \{1, 2, \dots, n\}, \quad (3)$$

$$0 \leq x_i \leq \min\{u_i, d\} \quad \text{for } 1 \leq i \leq m, \quad (4)$$

$$\text{There is no direct cycle in } x, \quad (5)$$

$$C(x) \leq c, \quad (6)$$

where (v, \cdot) is the set of arcs emanating from node v and (\cdot, v) is the set of arcs pointing to node v . It should be pointed out that (2)–(4) are established on the basis of the well-known flow conservation law [9]. By lemma, it is easy to obtain the following conclusion.

Corollary 2. For a state vector x with $M(x) = d$, if x satisfies conditions (3), (4), (5), and (6), then x is a (d, c) -MP.

By Lemma 1, the enumeration method can be used to search for (d, c) -MPs. Given a multistate network $G(V, E, u, C)$, (4) shows that the total number of state vectors contained in search space of (d, c) -MPs is $\prod_{i=1}^m (\min\{u_i, d\} + 1)$ which is a pretty huge number. Thus, it is inefficient to search for (d, c) -MPs by Lemma 1. Below, we will introduce how to divide the search space into disjoint subspaces from which all (d, c) -MPs can be efficiently obtained.

3. The Proposed Algorithm

Given a set of state vectors $X = \{x \mid x = (x_1, x_2, \dots, x_m), x_i \in K_i^X \text{ for } 1 \leq i \leq m\}$, where K_i^X is the set of states of e_i with respect to X , that is, $K_i^X = \{x_i \mid l_i^X \leq x_i \leq u_i^X \text{ for } 1 \leq i \leq m\}$, if we limit the search space of (d, c) -MP in set X , it is apparent that Lemma 1 can be transformed into the following conclusion.

Corollary 3. A state vector x is a (d, c) -MP in set X if and only if x satisfies the following conditions:

$$\sum_{e_i \in (s, \cdot)} x_i = \sum_{e_j \in (\cdot, t)} x_j = d, \quad (7)$$

$$\sum_{e_i \in (\cdot, v)} x_i = \sum_{e_j \in (v, \cdot)} x_j \quad (8)$$

for every node $v \in \{1, 2, \dots, n\}$,

$$l_i^X \leq x_i \leq \min\{u_i^X, d\} \quad \text{for } 1 \leq i \leq m, \quad (9)$$

$$\text{There is no direct cycle in } x, \quad (10)$$

$$C(x) \leq c. \quad (11)$$

It should be noted that the only difference between Lemma 1 and Corollary 3 is the state range of e_i . Next, we will consider a special state vector in set X : the smallest state vector l^X . It is clear that there are three cases with respect to $M(l^X)$: (1) Case 1: $M(l^X) > d$; (2) Case 2: $M(l^X) = d$; and (3) Case 3: $M(l^X) < d$.

Case 1 ($M(l^X) > d$). If $M(l^X) > d$, we have the following statement.

Theorem 4. For set X , if $M(l^X) > d$, there is no (d, c) -MP in X .

Proof. Since l^X is the smallest state vector in X , we have $x \geq l^X$ for any vector $x \in X$. If $M(l^X) > d$, then $M(x) \geq M(l^X) > d$ for any $x \in X$. That is, $M(x) > d$ for any $x \in X$; then, by the definition of (d, c) -MP, there is no (d, c) -MP in X . \square

According to Theorem 4, once we have verified that $M(l^X) > d$ holds, set X will be discarded (no (d, c) -MP exists in X). Thus, the search space of (d, c) -MPs will be reduced, which is extremely beneficial in solving (d, c) -MPs.

Case 2 ($M(l^X) = d$). In such case, we can claim that the following theorem holds.

Theorem 5. For set X , if $M(l^X) = d$, there exists no more than one (d, c) -MP in set X . Moreover, if there exists one (d, c) -MP, l^X is the unique (d, c) -MP in set X .

Proof. Suppose vector x ($\neq l^X$) is a (d, c) -MP in set X ; then $M(x) = d$. Note that $x > l^X$, and $M(x) = M(l^X) = d$, which is contrary to the definition of (d, c) -MP. Therefore, only l^X has the chance to be a (d, c) -MP in X . \square

Theorem 5 provides an effective method to search for (d, c) -MPs in set X . If $M(l^X) = d$ holds, we only need to check whether l^X is a (d, c) -MP by Corollary 3. If l^X is a (d, c) -MP, it is the unique (d, c) -MP in set X . Otherwise, there is no (d, c) -MP in set X .

Case 3 ($M(l^X) < d$). In such case, we need to analyze the largest state vector u^X in set X . If $M(u^X) < d$, we have the following statement.

Theorem 6. For set X , if $M(u^X) < d$, there is no (d, c) -MP in set X .

Proof. Since u^X is the largest state vector in set X , we have $x \leq u^X$ for any $x \in X$. If $M(u^X) < d$, it is clear to derive $M(x) \leq M(u^X) < d$ for any $x \in X$. That is, $M(x) < d$ for any $x \in X$; then, by the definition of (d, c) -MP, there is no d -MP in set X . \square

It is noteworthy that all of the three theorems contribute to reducing the search space of (d, c) -MPs. If $M(u^X) \geq d$, then we have $M(l^X) < d \leq M(u^X)$. In such case, the enumeration algorithm can be used to search for (d, c) -MPs in set X by Corollary 3. Now we consider a special case wherein set X satisfies $l^X = (0, 0, \dots, 0)$ and $M(u^X) \geq d$ simultaneously. The following theorem shows that a d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ can be derived from X using the max-flow algorithm, where d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ consists of flows f_i^d through each arc e_i for $1 \leq i \leq m$ when the max-flow of the network from s to t is d .

Theorem 7. For set X , if $l^X = (0, 0, \dots, 0)$ and $M(u^X) \geq d$, then there exists at least one d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ such that $(f_1^d, f_2^d, \dots, f_m^d) \in X$.

Proof. Since $M(u^X) \geq d$, at least d units of flow can be transmitted from the source to the destination when the network is under u^X . Also, such a d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ can be obtained by the following steps [10, 12, 13]:

- (a) Add a fictitious node t^* , and add an arc with a fixed state d from t to t^* .
- (b) The state of each arc $e_i \in E$ is defined by u_i^X .
- (c) Determine the max-flow from s to t^* . Since $M(u^X) \geq d$, and the state of the arc from t to t^* is d , the max-flow from s to t^* is d .
- (d) For each $e_i \in E$, the flows f_i^d through e_i constitute a d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ whenever the max-flow d from s to t^* is determined.

Since the flows f_i^d through e_i are upper bounded by u_i^X , we have $f_i^d \leq u_i^X$ for all $e_i \in E$. Meanwhile, it is apparent that $f_i^d \geq 0$ for all $e_i \in E$. Thus, we have $0 \leq f_i^d \leq u_i^X$ for all $e_i \in E$; that is, $l_i^X \leq f_i^d \leq u_i^X$ for all $e_i \in E$. Consequently, we have d -flow $(f_1^d, f_2^d, \dots, f_m^d) \in X$. \square

The definition of d -flow indicates that a d -flow satisfies the flow conservation law; thus, it satisfies (7)–(9). Accordingly, for such a d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ derived from X by Theorem 7, if there is no directed cycle in it and $\sum_{i=1}^m C_i f_i^d \leq c$, it is a (d, c) -MP. Otherwise, it is not a (d, c) -MP. Grounded on the obtained d -flow $(f_1^d, f_2^d, \dots, f_m^d)$, set X can be decomposed into disjoint subsets from which other (d, c) -MPs can be derived. The decomposition technique which is proposed by Jane and Lai [13] will be described below.

Let $E^d(X) = \{e_i \mid e_i \in E \ \& \ f_i^d > 0\}$; then it is obvious that $E^d(X)$ is nonempty. Suppose $E^d(X) = \{e_{z_1}, e_{z_2}, \dots, e_{z_q}\}$, by pivoting on arcs $e_{z_1}, e_{z_2}, \dots, e_{z_q}$ one by one; X is decomposed into $q + 1$ nonempty disjoint subsets X^1, X^2, \dots, X^q , and X^0 as follows:

- (1) Pivot on arc e_{z_1} :

$$X^1 = \{(x_1, x_2, \dots, x_m) \mid 0 \leq x_{z_1} \leq (f_{z_1}^d - 1), \ x_i \in K_i^X \text{ for } e_i \in E \setminus \{e_{z_1}\}\}. \quad (12)$$

- (2) Pivot on arc e_{z_2} :

$$X^2 = \{(x_1, x_2, \dots, x_m) \mid f_{z_1}^d \leq x_{z_1} \leq u_{z_1}^X, \ 0 \leq x_{z_2} \leq (f_{z_2}^d - 1), \ x_i \in K_i^X \text{ for } e_i \in E \setminus \{e_{z_1}, e_{z_2}\}\}. \quad (13)$$

- (3) :

- (q) Pivot on arc e_{z_q} : $X^q = \{(x_1, x_2, \dots, x_m) \mid f_i^d \leq x_i \leq u_i^X \text{ for } e_i \in \{e_{z_1}, e_{z_2}, \dots, e_{z_{q-1}}\}, \ 0 \leq x_{z_q} \leq (f_{z_q}^d - 1), \ x_i \in K_i^X \text{ for } e_i \in E \setminus E^d(X)\}$

- (q + 1) $X^0 = \{(x_1, x_2, \dots, x_m) \mid f_i^d \leq x_i \leq u_i^X \text{ for } e_i \in E^d(X), \ x_i \in K_i^X \text{ for } e_i \in E \setminus E^d(X)\}$.

Theorem 8. (i) $X = X^1 \cup X^2 \cup \dots \cup X^q \cup X^0$; (ii) subsets X^1, X^2, \dots, X^q , and X^0 are nonempty and disjoint; (iii) d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ is the smallest state vector in X^0 .

Readers may refer to the work of Jane and Laih [13] for the proofs. Now, we first consider subset X^0 . Since the d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ derived from Theorem 7 is the smallest state vector in set X^0 , if it is a (d, c) -MP, it is the unique (d, c) -MP in set X^0 ; otherwise, there is no (d, c) -MP in set X^0 . As a result, there is no need to check other state vectors in X^0 ; that is, X^0 can be discarded as soon as d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ has been checked, which will reduce the number of enumerated state vectors to a large extent.

Then, we discuss subsets X^i ($2 \leq i \leq q$). For each X^i ($2 \leq i \leq q$), if $M(u^{X^i}) < d$ or $M(l^{X^i}) > d$ for subset X^i ($2 \leq i \leq q$), there is no (d, c) -MP in X^i by Theorem 6 or Theorem 4. Then X^i is discarded, which will also cut down the search space of (d, c) -MPs. If $M(l^{X^i}) = d$, we only need to check whether l^{X^i} is a (d, c) -MP by Theorem 5. Otherwise, we have $M(l^{X^i}) < d \leq M(u^{X^i})$, and thus the enumeration algorithm can be employed to search for (d, c) -MPs in X^i by Corollary 3.

Finally, we discuss subset X^1 . It is noted that $l^{X^1} = (0, 0, \dots, 0)$. If $M(u^{X^1}) < d$, X^1 is also discarded. Otherwise, subset X^1 satisfies $l^X = (0, 0, \dots, 0)$ and $M(u^{X^1}) \geq d$ simultaneously. Consequently, Theorem 7 can be applied to obtain a d -flow from X^1 . Based on the derived d -flow, X^1 is decomposed into nonempty disjoint subsets in the same manner. As discussed earlier, all of these subsets can be handled similarly. The decomposition procedure will be terminated when no subset, \underline{X} , satisfies $l^{\underline{X}} = (0, 0, \dots, 0)$ and $M(u^{\underline{X}}) \geq d$ simultaneously.

Based on the above discussions, all (d, c) -MPs can be found using the following steps.

Step 0. Let $X = \{x \mid x = (x_1, x_2, \dots, x_m), 0 \leq x_i \leq \min\{u_i, d\}$ for $1 \leq i \leq m\}$.

Step 1. Compute $M(u^X)$. If $M(u^X) < d$, there is no (d, c) -MP in X (by Theorem 6), and halt.

Step 2. Seek a d -flow $(f_1^d, f_2^d, \dots, f_m^d)$ by Theorem 7 and compute $E^d(X)$.

Step 3. For $e_i \in E^d(X)$, X is decomposed into X^1, X^2, \dots, X^q , and X^0 by Theorem 8.

Step 4. For X^0 , if there is no directed cycle in it and $\sum_{i=1}^m C_i f_i^d \leq c$, it is a (d, c) -MP; otherwise, it is not a (d, c) -MP.

Step 5. For $i = 2$ to q ,

(5.1) compute $M(u^{X^i})$; if $M(u^{X^i}) < d$, there is no (d, c) -MP in X^i (by Theorem 6), and go to (5.4);

(5.2) compute $M(l^{X^i})$; if $M(l^{X^i}) > d$, there is no (d, c) -MP in X^i (by Theorem 4); if $M(l^{X^i}) = d$, check if l^{X^i} is a (d, c) -MP by Corollary 3, and go to (5.4);

(5.3) use the enumeration algorithm to search for (d, c) -MPs in X^i by Corollary 3;

(5.4) $i = i + 1$.

Step 6. Let $X = X^1$, and go to Step 1.

The time complexity of the proposed method is discussed below. Either Step 0 or Step 6 takes $O(m)$ time. Step 1 takes $O(n^3)$ to evaluate $M(u^X)$ [14]. A d -flow can be found in $O(n^3)$ time by the max-flow algorithm. It requires $O(m)$ time to obtain $E^d(X)$ [13]. Then, Step 2 requires $O(n^3)$ time in total. Step 3 takes $O(m)$ time to decompose X . Step 4 requires $O(m + n)$ time. Therefore, the algorithm totally takes $O(n^3)$ time from Steps 0 to 4. It is apparent that the time complexity of Step 5 is determined by the number of enumerated state vectors. The number of state vectors in Step 5 is less than $\prod_{i=1}^m (\min\{u_i, d\} + 1)$; thus, the time complexity of Step 5 is less than $O((m + n) \prod_{i=1}^m (\min\{u_i, d\} + 1))$. Therefore, the time complexity of the proposed algorithm is less than $O((m + n) \prod_{i=1}^m (\min\{u_i, d\} + 1))$.

4. An Illustrative Example

In this section, we use a simple logistics network to demonstrate the proposed algorithm step by step. The network, as shown in Figure 1, is cited from the literature in [6–9]. The data of each arc is given in Table 1. If the manager is assigned a budget of 14 ($c = 14$) to transport 3 units of commodity (demand level $d = 3$) from s to t , the reliability index $R_{(3,14)}$ can be obtained as follows (for convenience, the set of state vectors X is denoted by its smallest and largest state vectors as $[l^X, u^X]$).

Step 0. Let $X = [(0, 0, 0, 0, 0, 0), (3, 2, 1, 1, 2, 2)]$.

Step 1. $M(u^X) = 4 > 3$.

Step 2. Seek a 3-flow $(3, 2, 1, 0, 0, 1)$, and compute $E^3(X) = \{e_1, e_2, e_3, e_6\}$.

Step 3. X is decomposed into X^1, X^2, X^3, X^4 , and X^0 , where $X^1 = [(0, 0, 0, 0, 0, 0), (2, 2, 1, 1, 2, 2)]$, $X^2 = [(3, 0, 0, 0, 0, 0), (3, 1, 1, 1, 2, 2)]$, $X^3 = [(3, 2, 0, 0, 0, 0), (3, 2, 0, 1, 2, 2)]$, $X^4 = [(3, 2, 1, 0, 0, 0), (3, 2, 1, 1, 2, 0)]$, and $X^0 = [(3, 2, 1, 0, 0, 1), (3, 2, 1, 1, 2, 2)]$.

Step 4. Since $C(l^{X^0}) = 15 > 14$, $l^{X^0} = (3, 2, 1, 0, 0, 1)$ is not a $(3, 14)$ -MP.

Step 5. $i = 2$.

(5.1) $X^2 = [(3, 0, 0, 0, 0, 0), (3, 1, 1, 1, 2, 2)]$, and $M(u^{X^2}) = 3$.

(5.2) $M(l^{X^2}) = 0$.

(5.3) Use the enumeration algorithm to search for (3, 14)-MPs in X^2 , but there is no (3, 14)-MP.

(5.4) $i = 3$.

(5.1) $X^3 = [(3, 2, 0, 0, 0, 0), (3, 2, 0, 1, 2, 2)]$, and $M(u^{X^3}) = 4$.

(5.2) $M(l^{X^3}) = 2$.

(5.3) Use the enumeration algorithm to search for (3, 14)-MPs in X^3 , but there is no (3, 14)-MP.

(5.4) $i = 4$.

(5.1) $X^4 = [(3, 2, 1, 0, 0, 0), (3, 2, 1, 1, 2, 0)]$, and $M(u^{X^4}) = 2 < 3$; then there is no (3, 14)-MP in X^4 .

Step 6. Let $X = X^1 = [(0, 0, 0, 0, 0, 0), (2, 2, 1, 1, 2, 2)]$, and go to Step 1.

Step 1. $M(u^X) = 4 > 3$.

Step 2. Seek a 3-flow $(2, 2, 0, 0, 1, 1)$, $E^3(X) = \{e_1, e_2, e_5, e_6\}$.

Step 3. X is decomposed into X^{1-1} , X^{1-2} , X^{1-3} , X^{1-4} , and X^{1-0} , where $X^{1-1} = [(0, 0, 0, 0, 0, 0), (1, 2, 1, 1, 2, 2)]$, $X^{1-2} = [(2, 0, 0, 0, 0, 0), (2, 1, 1, 1, 2, 2)]$, $X^{1-3} = [(2, 2, 0, 0, 0, 0), (2, 2, 1, 1, 0, 2)]$, $X^{1-4} = [(2, 2, 0, 0, 1, 0), (2, 2, 1, 1, 2, 0)]$, and $X^{1-0} = [(2, 2, 0, 0, 1, 1), (2, 2, 1, 1, 2, 2)]$.

// the disjoint sets of state vectors are denoted here as X^{1-1} , X^{1-2} , X^{1-3} , X^{1-4} , and X^{1-0} //

Step 4. Since there is no directed cycle in $(2, 2, 0, 0, 1, 1)$, and $C(l^{X^{1-0}}) = 12 < 14$, $l^{X^{1-0}} = (2, 2, 0, 0, 1, 1)$ is a (3, 14)-MP.

Step 5. $i = 2$.

(5.1) $X^{1-2} = [(2, 0, 0, 0, 0, 0), (2, 1, 1, 1, 2, 2)]$, and $M(u^{X^{1-2}}) = 3$.

(5.2) $M(l^{X^{1-2}}) = 0$.

(5.3) Use the enumeration algorithm to search for (3, 14)-MPs in X^{1-2} , but there is no (3, 14)-MP.

(5.4) $i = 3$.

(5.1) $X^{1-3} = [(2, 2, 0, 0, 0, 0), (2, 2, 1, 1, 0, 2)]$, and $M(u^{X^{1-3}}) = 2 < 3$, then there is no (3, 14)-MP in X^{1-3} , and go to (5.4).

(5.4) $i = 4$.

(5.1) $X^{1-4} = [(2, 2, 0, 0, 1, 0), (2, 2, 1, 1, 2, 0)]$, and $M(u^{X^{1-4}}) = 2 < 3$; then there is no (3, 14)-MP in X^{1-4} .

Step 6. Let $X = X^{1-1} = [(0, 0, 0, 0, 0, 0), (1, 2, 1, 1, 2, 2)]$, and go to Step 1.

Step 1. $M(u^X) = 3$.

Step 2. Seek a 3-flow $(1, 1, 0, 0, 2, 2)$, $E^3(X) = \{e_1, e_2, e_5, e_6\}$.

Step 3. X is decomposed into X^{1-1-1} , X^{1-1-2} , X^{1-1-3} , X^{1-1-4} , and X^{1-1-0} , where $X^{1-1-1} = [(0, 0, 0, 0, 0, 0), (0, 2, 1, 1, 2, 2)]$, $X^{1-1-2} = [(1, 0, 0, 0, 0, 0), (1, 0, 1, 1, 2, 2)]$, $X^{1-1-3} = [(1, 1, 0, 0, 0, 0), (1, 2, 1, 1, 1, 2)]$, $X^{1-1-4} = [(1, 1, 0, 0, 2, 0), (1, 2, 1, 1, 2, 1)]$, and $X^{1-1-0} = [(1, 1, 0, 0, 2, 2), (1, 2, 1, 1, 2, 2)]$.

//the disjoint sets of state vectors are denoted here as X^{1-1-1} , X^{1-1-2} , X^{1-1-3} , X^{1-1-4} , and X^{1-1-0} //

Step 4. Since there is no directed cycle in $(1, 1, 0, 0, 2, 2)$, and $C(l^{X^{1-1-0}}) = 12 < 14$, $l^{X^{1-1-0}} = (1, 1, 0, 0, 2, 2)$ is a (3, 14)-MP.

Step 5. $i = 2$.

(5.1) $X^{1-1-2} = [(1, 0, 0, 0, 0, 0), (1, 0, 1, 1, 2, 2)]$, and $M(u^{X^{1-1-2}}) = 2$, then there is no (3, 14)-MP in X^{1-1-2} , and go to (5.4).

(5.4) $i = 3$.

(5.1) $X^{1-1-3} = [(1, 1, 0, 0, 0, 0), (1, 2, 1, 1, 1, 2)]$, and $M(u^{X^{1-1-3}}) = 2$, then there is no (3, 14)-MP in X^{1-1-3} , and go to (5.4).

(5.4) $i = 4$.

(5.1) $X^{1-1-4} = [(1, 1, 0, 0, 2, 0), (1, 2, 1, 1, 2, 1)]$, and $M(u^{X^{1-1-4}}) = 3$.

(5.2) $M(l^{X^{1-1-4}}) = 1$.

(5.3) Use the enumeration algorithm to search for (3, 14)-MPs in X^{1-1-4} , and $(1, 2, 0, 1, 2, 1)$ is a (3, 14)-MP.

Step 6. Let $X = X^{1-1-1} = [(0, 0, 0, 0, 0, 0), (0, 2, 1, 1, 2, 2)]$, and go to Step 1.

Step 1. $M(u^X) = 2 < 3$, and halt.

Consequently, $(2, 2, 0, 0, 1, 1)$, $(1, 1, 0, 0, 2, 2)$, and $(1, 2, 0, 1, 2, 1)$ are all (3, 14)-MPs. Let $B_1 = \{x \mid x \geq (2, 2, 0, 0, 1, 1)\}$, $B_2 = \{x \mid x \geq (1, 2, 0, 1, 2, 1)\}$, and $B_3 = \{x \mid x \geq (1, 1, 0, 0, 2, 2)\}$; then, according to (1), $R_{(3,14)} = \Pr\{B_1 \cup B_2 \cup B_3\} = \Pr\{B_1\} + \Pr\{B_2\} + \Pr\{B_3\} - \Pr\{B_1 \cap B_2\} - \Pr\{B_1 \cap B_3\} - \Pr\{B_2 \cap B_3\} + \Pr\{B_1 \cap B_2 \cap B_3\} = \Pr\{x \mid x \geq (2, 2, 0, 0, 1, 1)\} + \Pr\{x \mid x \geq (1, 2, 0, 1, 2, 1)\} + \Pr\{x \mid x \geq (1, 1, 0, 0, 2, 2)\} - \Pr\{x \mid x \geq (2, 2, 0, 1, 2, 1)\} - \Pr\{x \mid x \geq (2, 2, 0, 0, 2, 2)\} - \Pr\{x \mid x \geq (1, 2, 0, 1, 2, 2)\} + \Pr\{x \mid x \geq (2, 2, 0, 1, 2, 2)\} = 0.64005$.

That is, the probability of transmitting 3 units of commodity demand from s to t with the total transportation cost less than or equal to 14 is 0.64005 (note that the final result $R_{(3,14)} = 0.66861$ in [7, 9] is incorrect). The value, 0.64005, gives the manager information on the capability of the network to ensure the shipment of 3 units of commodity within the budget 14 from s to t . If this value is below the threshold set by the manager, more investments are needed to improve the logistics network. Otherwise, the performance of the logistics network is desirable.

5. Computational Experiments

As pointed out in Section 2, the algorithms of Yeh [8] and Niu and Xu [9] solve the (d, c) -MP problem without

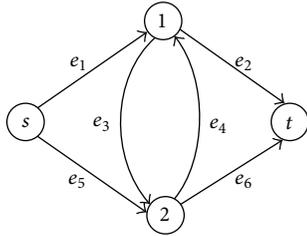


FIGURE 2

TABLE 2: The computational time of different algorithms.

(d, c) -MP	Computational time (s)			Relative ratio	
	T_{Yeh}	$T_{\text{Niu\&Xu}}$	T_{New}	$T_{\text{Yeh}}/T_{\text{New}}$	$T_{\text{Niu\&Xu}}/T_{\text{New}}$
(1, 10)-MP	0.142	0.721	0.138	1.029	5.225
(2, 17)-MP	5.103	5.799	0.669	7.628	8.668
(3, 27)-MP	14.275	9.705	1.101	12.965	8.815
(4, 36)-MP	20.841	5.752	1.197	17.411	4.805
(5, 49)-MP	20.319	0.831	0.814	24.451	1.021
(6, 60)-MP	19.409	0.627	0.472	41.121	1.328

knowing all MPs in advance. Therefore, we conduct numerical experiments to check the performance of the proposed algorithm by comparing it with the algorithms of Yeh and Niu and Xu. A multistate network, as shown in Figure 2, is cited from the literature in [9]. The largest state vector and the cost vector are $u = (2, 4, 2, 2, 1, 3, 2, 3, 1, 2, 2, 2, 4)$ and $C = (3, 2, 1, 2, 1, 2, 3, 2, 1, 3, 1, 2, 3)$, respectively. All of the three algorithms are implemented in a MATLAB program with a PC (AMD Athlon (tm) 64 X2 1.71 GHz CPU). To comprehensively embody the performance of the three algorithms, six different combinations of demand level and cost constraint are randomly chosen. The computational results are listed in Table 2.

It is observed from Table 2 that the proposed algorithm spends less time than Yeh's [8] in searching for different (d, c) -MPs. In particular, the proposed algorithm has a better searching efficiency as the demand level d grows. This is as expected, because Yeh's algorithm is an exhaustive enumeration method in nature, whereas the proposed algorithm utilizes a decomposition technique to divide the search space, such that the search space of (d, c) -MPs can be dramatically reduced, and thereby (d, c) -MPs can be efficiently obtained. Furthermore, Table 2 shows that the proposed algorithm displays an advantage over the algorithm of Niu and Xu [9]. Both of them are partial-enumeration methods, but the proposed technique is more effective in shortening the search space of (d, c) -MPs. Therefore, the proposed algorithm outperforms the two methods in solving (d, c) -MPs.

6. Conclusions

Reliability analysis is a fundamental tool for understanding the operational level of logistics networks. Logistics network can be modeled as a typical multistate flow network in which arcs are associated with multiple integer capacities,

operational reliability, and unit transportation cost. The reliability index $R_{(d,c)}$ which takes into account both reliability and transportation cost can serve as a comprehensive performance assessment of logistics networks. This paper is devoted to proposing a new efficient algorithm for solving all (d, c) -MPs. A major contribution of this paper is that we develop a new approach to shorten the search space of (d, c) -MP. Specifically, the proposed algorithm employs an effective decomposition technique to divide the search space of (d, c) -MPs such that the search space of (d, c) -MPs can be dramatically reduced. A real example is used to illustrate the applicability of the proposed algorithm. Also, to check the performance of the proposed algorithm, computational experiments on a medium-sized logistics network are conducted, and the results show that the proposed algorithm has an advantage over the existing methods [5–9].

For future research, there is still a room for developing efficient methods for $R_{(d,c)}$. For example, as with the existing algorithms [4–9], the proposed algorithm is also an enumeration method in nature, and thus it still needs to perform the enumeration of state vectors. Therefore, it is more desirable to develop new methods which are not grounded on the enumeration of state vectors. Furthermore, we only consider the transmission of single type of commodity in the network. Thus, it is more practical to extend the proposed algorithm to the reliability evaluation of multicommodity multistate networks under cost constraint in which multiple types of commodity are transmitting from the source to the destination.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (61300124 and 61403128), the Key Project of Science and Technology Research from the Education Department of Henan Province (13B120022 and 13B630034), and the Young Foundation of Henan Polytechnic University (Q2014-09).

References

- [1] Y.-K. Lin, "System reliability evaluation for a multistate supply chain network with failure nodes using minimal paths," *IEEE Transactions on Reliability*, vol. 58, no. 1, pp. 34–40, 2009.
- [2] Y.-K. Lin, "Performance evaluation for the logistics system in case that capacity weight varies from arcs and types of commodity," *International Journal of Production Economics*, vol. 107, no. 2, pp. 572–580, 2007.
- [3] C.-C. Jane, "Performance evaluation of logistics systems under cost and reliability considerations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 130–137, 2011.
- [4] C.-C. Jane and Y.-W. Laih, "Evaluating cost and reliability integrated performance of stochastic logistics systems," *Naval Research Logistics*, vol. 59, no. 7, pp. 577–586, 2012.

- [5] J.-S. Lin, "Reliability evaluation of capacitated-flow networks with budget constraints," *IIE Transactions*, vol. 30, no. 12, pp. 1175–1180, 1998.
- [6] Y.-K. Lin, "Reliability of a stochastic-flow network with unreliable branches & nodes, under budget constraints," *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 381–387, 2004.
- [7] W.-C. Yeh, "A new approach to evaluate reliability of multistate networks under the cost constraint," *Omega*, vol. 33, no. 3, pp. 203–209, 2005.
- [8] W. C. Yeh, "An improved method for multistate flow network reliability with unreliable nodes and a budget constraint based on path set," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 41, no. 2, pp. 350–355, 2011.
- [9] Y.-F. Niu and X.-Z. Xu, "Reliability evaluation of multi-state systems under cost consideration," *Applied Mathematical Modelling*, vol. 36, no. 9, pp. 4261–4270, 2012.
- [10] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 1962.
- [11] W.-C. Yeh, "A novel method for the network reliability in terms of capacitated-minimum-paths without knowing minimum-paths in advance," *Journal of the Operational Research Society*, vol. 56, no. 10, pp. 1235–1240, 2005.
- [12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, USA, 1993.
- [13] C.-C. Jane and Y.-W. Laih, "A practical algorithm for computing multi-state two-terminal reliability," *IEEE Transactions on Reliability*, vol. 57, no. 2, pp. 295–302, 2008.
- [14] R. K. Ahuja, M. Kodialam, A. K. Mishra, and J. B. Orlin, "Computational investigations of maximum flow algorithms," *European Journal of Operational Research*, vol. 97, no. 3, pp. 509–542, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

