Transportation Research Part E 100 (2017) 75–97

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Evaluating the reliability of a stochastic distribution network in terms of minimal cuts

Yi-Feng Niu [a,b], Zi-You Gao [a,*], William H.K. Lam [a,c]

[a] School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
[b] School of Mathematics & Information Science, Henan Polytechnic University, Jiaozuo 454000, China
[c] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 8 September 2016
Received in revised form 29 January 2017
Accepted 31 January 2017

Keywords:
Reliability
Stochastic distribution network
Minimal cut
D-minimal cut

ABSTRACT

This paper presents a d-minimal cut based algorithm to evaluate the performance index R_{d+1} of a distribution network, defined as the probability that a specified demand $d+1$ can be successfully distributed through stochastic arc capacities from the source to the destination. To improve the efficiency of solving d-minimal cuts, a novel technique is developed to determine the minimal capacities of arcs. Also, two new judging criteria are proposed to detect duplicate d-minimal cuts. Both theoretical and computational results indicate that our algorithm outperforms the existing methods. Furthermore, a real case study is provided to illustrate the application of the algorithm.

1. Introduction

1.1. Background

Logistics distribution networks provide the infrastructure for the storage and distribution of products. In the context of either general business logistics (Chopra, 2003; Sheu, 2006) or emergency logistics (Edrissi et al., 2015; Sheu, 2007, 2010), distribution activity is considered as the process of the transfer of products from supply points to demand points. Relative to other logistics functions, such as procurement, manufacturing, warehousing, inventory and information systems, distribution is a key function in the entire logistics system and the key link between manufacturers and customers in a supply chain (Yang, 2013). Furthermore, distribution is a major driver of profitability in a company due to its direct impact on both the logistics cost and the customer experience (Chopra, 2003). Therefore, a distribution network with better performance plays a significant role in achieving a number of logistics and supply chain goals, ranging from low operational cost to high customer service level (Chopra, 2003; Ho and Emrouznejad, 2009; Peng et al., 2011; Tsao and Lu, 2012; Whicker et al., 2009; Yang, 2013).

The performance evaluation of distribution networks is a popular issue in the field of logistics and supply chain management. Chopra (2003) pointed out that at the highest level, the performance of a distribution network can be evaluated along two dimensions: meeting customer needs, and cost of meeting customer needs. Also, many researchers have studied the performance evaluation of distribution networks according to the following questions: "Have customer demands been fulfilled?" "Is the total cost minimized?" and "Have products been timely delivered?" in which several important factors

* Corresponding author.
E-mail address: zygao@bjtu.edu.cn (Z.-Y. Gao).

http://dx.doi.org/10.1016/j.tre.2017.01.008
1366-5545/

affecting the performance are considered, such as cost, service level, lead time, product availability, transportation capacity, or market demand (Ho and Emrouznejad, 2009; Nagurney et al., 2014, 2015; Tsao and Lu, 2012; Whicker et al., 2009; Yu and Nagurney, 2013). Of note is that the distribution networks addressed in the aforementioned studies are deterministic. In practical applications, deterministic models fail to fully characterize the actual performance of a distribution network that is always subject to many types of uncertainty (Soltani-Sobh et al., 2015, 2016a). Lin et al. (2013) and Yeh et al. (2014) argued that any distribution network can be regarded as a typical stochastic-flow network (also called multi-state network), and assessing the performance of distribution networks in uncertain states is of crucial importance to maintain a high level of operation in the whole logistics system (Lin, 2007, 2009; Niu et al., 2014; Jane, 2011).

A distribution network can be represented as both sets of nodes and arcs, where each node stands for a supplier, a transfer center, or a market (e.g., a wholesaler, a retailer, or a customer), and each arc (or called route, link) connecting a pair of nodes stands for an air route, a land route, or an ocean route. Along each arc, there is a carrier to provide the transportation service. Owing to the effect of unexpected situation in reality, the available capacity of each carrier is stochastic (Lin et al., 2013; Yeh et al., 2014). For example, the vehicles owned by one carrier may be in a failure state, partial failure state, or maintenance state, such that the number of vehicles available is stochastic. In that sense, each arc has several random capacities that can be described with a probability distribution. And, the goods transported through such a distribution network are reckoned as a flow. For a distribution network with random arc capacities, the network capacity (the maximum flow from the source to the destination) is not a fixed value, so whether the network can successfully deliver sufficient amount of commodity to meet market demand is not a simple yes or no question. In such a case, reliability analysis can serve as a useful tool to measure the network performance.

1.2. Network reliability

Reliability is a fundamental attribute for the safe operation of any modern technological system, and is generally defined as the probability that a system performs its intended function within a given time horizon and environment (Zio, 2009; Peng et al., 2011). This definition is particularly focused on the situation in which components of the system may fail or partially fail due to a variety of uncertainties during operation. Traditionally, network reliability study has been centered mainly on three aspects (Soltani-Sobh et al., 2016a, 2016b; Chen et al., 2002, 2013; Cedillo-Campos et al., 2014): (i) connectivity reliability—the probability that the nodes of the network remain connected; (ii) travel time reliability—the probability that a successful travel from the source to the destination can be made within a specified interval of time; and (iii) capacity reliability—probability that a specified flow demand can be successfully transported from the source to the destination. In addition to the above-mentioned three types, research has also been dedicated to other reliability measures. For instance, the study by Soltani-Sobh et al. (2016a) is focused on behavioral reliability by considering the uncertainty in people's travel making decision, where behavioral reliability is concerned with the effect of the modified mean behavior of travelers on the mean network performance. Soltani-Sobh et al. (2016b) utilized performance reliability, defined as the probability that the performance measure as a function of random variables are in the safe region and acceptable level, to analyze a transportation network subject to unexpected events with multiple uncertainties. Among these reliability measures, capacity reliability which combines the source–destination connection, arc capacity constraint and flow demand is the most commonly employed indicator to assess the performance of many real-world systems, and is the focus of this paper.

1.3. Capacity reliability evaluation

Reliability evaluation has been shown to be an NP-hard problem (Ball, 1993; Colbourn, 1987), although it has been extensively studied. Common in the literature is the two-terminal capacity reliability (2TCR), a classical reliability index with a broad range of practical applications (Ramirez-Marquez and Coit, 2005b). Given a stochastic-flow network whose components take discrete, non-negative integer values following a certain probability distribution, two-terminal capacity reliability at demand level $d+1$ ($2TCR_{d+1}$) is defined as the probability that $d+1$ units of flow demand can be successfully distributed from the source to the destination. Virtually, $2TCR_{d+1}$ can be looked upon as a combination of the source–destination delivery, arc capacity, and flow demand (Jane, 2011).

From the perspective of reliability evaluation, a great deal of research (Alexopoulos, 1995; Doulliez and Jamoulle, 1972; Jane and Laih, 2008, 2010; Jane et al., 1993; Lin, 2002; Yeh, 2002, 2004; Yan and Qian, 2007; Yeh, 2008; Forghani-elahabad and Mahdavi-Amiri, 2014; Yeh et al., 2015) has been devoted to calculating $2TCR_{d+1}$. The algorithms in these studies can be broadly categorized as direct and indirect methods (Jane and Laih, 2008). The complete enumeration method solves $2TCR_{d+1}$ in a simple, and straightforward manner. It enumerates all possible combinations of arc states, so it is computationally expensive. The popular decomposition method for $2TCR_{d+1}$ is proposed by Doulliez and Jamoulle (1972). However, Alexopoulos (1995) pointed out that this direct decomposition method may yield incorrect results. Recently, Jane and Laih (2008, 2010) proposed two decomposition algorithms for the straightforward computation of $2TCR_{d+1}$. Based on a special capacity vector, Jane and Laih's algorithms repeatedly apply a novel decomposition technique to divide the set of capacity vectors, such that all acceptable (unacceptable) capacity vectors which are capable (incapable) of transmitting the required flow demand from the source to the destination can be attained. As a result, $2TCR_{d+1}$ can be easily obtained by computing the probabilities of all acceptable (unacceptable) capacity vectors.

In recent years, a large number of indirect algorithms that solve $2TCR_{d+1}$ by way of a medium have also been developed. In particular, one general method for $2TCR_{d+1}$ is using minimal cuts (MCs) which are shown to be a powerful tool for reliability evaluation (Yeh, 2008). A cut is a set of arcs whose removal results in the disconnection of the source node and the destination node. An MC is a cut whose proper subset is no longer a cut (Lin, 1998; Yeh, 2008). Under the assumption that all MCs are known in advance, these methods are focused on developing efficient procedures for seeking *d*-minimal cuts (*d*-MCs) (Jane et al., 1993; Lin, 2002; Yeh, 2002, 2004; Yan and Qian, 2007; Yeh, 2008; Forghani-elahabad and Mahdavi-Amiri, 2014; Yeh et al., 2015). A *d*-MC, *X*, is a maximal capacity vector exactly meeting the demand level *d*, which means $M(X) = d$, and $M(Y) > d$ for any $Y > X$ (Lin, 2002). Given that all *d*-MCs have been found, there are several known methods available to compute $2TCR_{d+1}$, such as the Inclusion-Exclusion (IE) method (Lin, 2002; Yeh, 2004; Forghani-elahabad and Mahdavi-Amiri, 2014), or the Sum of Disjoint Products (SDP) method (Kuo and Zuo, 2003; Yeh, 2015; Zuo et al., 2007; Bai et al., 2015). Therefore, the efficient solution of *d*-MCs is critical to the evaluation of $2TCR_{d+1}$. In general, the existing algorithms for solving all *d*-MCs consist of three major steps (Yeh et al., 2015).

Step 1. Solve all *d*-MC candidates from MCs.
Step 2. Verify *d*-MC candidates to attain real *d*-MCs.
Step 3. Remove duplicate *d*-MCs.

Since each *d*-MC is also a *d*-MC candidate, the existing methods need to search for all *d*-MC candidates prior to determining *d*-MCs (Step 1). But, given that a *d*-MC candidate is not necessarily a *d*-MC, a verification procedure is required (Step 2). The set of *d*-MCs derived from Step 2 may contain duplicate *d*-MCs which simply add to the difficulty of reliability evaluation but do not influence the reliability value, and thus a step for removing duplicate *d*-MCs is necessary (Step 3).

A body of research has contributed to the solution of the *d*-MC problem. Jane et al. (1993) first introduced the concept of *d*-MC candidates, and proposed a mathematical model to solve all *d*-MC candidates using the implicit enumeration method. They also proved that all *d*-MCs can be obtained from *d*-MC candidates. Lin (2002) showed that the comparison method can be used to determine *d*-MCs from *d*-MC candidates and eliminate duplicate *d*-MCs simultaneously. With some improvements in calculating the max-flow value, Yeh (2002) proposed a new method to verify whether a *d*-MC candidate is a *d*-MC. Yeh (2004) further improved his method (Yeh, 2002) to consider how to eliminate duplicate *d*-MCs using the restriction method. Based on some new results, Yan and Qian (2007) discussed how to add some constraints to reduce the number of *d*-MC candidates during enumeration. Yeh (2008) proposed an algorithm to decrease the number of *d*-MC candidates, and found that unsaturated components are the key components for detecting duplicate *d*-MCs. Forghani-elahabad and Mahdavi-Amiri (2014) proposed an efficient comparison method to eliminate duplicate *d*-MCs. By associating a number with each *d*-MC, the method compares the associated numbers to detect duplicate *d*-MCs instead of comparing all *d*-MCs. More recently, Yeh et al. (2015) presented some new results, and put forward a method to detect duplicate *d*-MCs.

1.4. Contributions of this paper

The main objective of this paper is to propose a *d*-MC based algorithm for evaluating the capacity reliability of a stochastic distribution network. Specifically, this paper provides three major contributions to the existing literature: (i) a novel technique is developed to efficiently determine the minimal capacities, called lower capacity bounds herein, of arcs in *d*-MCs, so as to advance the efficiency of solving *d*-MCs; (ii) two judging criteria are proposed to correctly and effectively detect duplicate *d*-MCs; (iii) A new efficient algorithm is provided to solve all *d*-MCs. Each contribution is further explained in the following subsections:

1.4.1. A novel technique for determining lower capacity bounds of arcs in d-MCs

It has been shown that the cost of solving *d*-MCs is directly dependent on the number of *d*-MC candidates (Yan and Qian, 2007; Yeh, 2008) which is always enormous. Hence, reducing the number of *d*-MC candidates is undoubtedly the most cost-effective manner to advance the efficiency of solving *d*-MCs. The works of Yan and Qian (2007), and Yeh (2008) indicate that the concept of lower capacity bound can be utilized to decrease the number of *d*-MC candidates. With this in mind, we develop a novel technique to efficiently find lower capacity bounds of arcs which can, on the one hand, be used to determine some special *d*-MCs without the tedious verification, and can, on the other hand, serve as constraints to shorten the capacity range of arcs in solving *d*-MCs, and thus to reduce the number of *d*-MC candidates.

1.4.2. Two judging criteria to correctly and effectively detect duplicate d-MCs

A major difficulty for solving *d*-MCs is how to effectively and efficiently eliminate duplicate *d*-MCs. The restriction method fails to effectively remove all duplicate *d*-MCs (Yeh, 2008). The popular comparison method for deleting duplicate *d*-MCs is simple yet inefficient (Yeh et al., 2015). Furthermore, the comparison method (Lin, 2002; Yan and Qian, 2007; Forghani-elahabad and Mahdavi-Amiri, 2014) always ignores a fundamental issue why two distinct MCs can generate identical *d*-MCs. Note that a few papers, including Yeh (2008) and Yeh et al. (2015), have made an attempt to seek the reason for the generation of duplicate *d*-MCs, but the methods in these papers, i.e., the methods of Yeh (2008) and Yeh et al. (2015), fail to correctly detect duplicate *d*-MCs in some special cases. Therefore, there is a growing demand for developing new approaches to detect duplicate *d*-MCs. In this paper, we propose two judging criteria to correctly and effectively detect duplicate d-MCs. The two criteria not only provide efficient approaches to remove duplicate d-MCs, but also find out the underlying reason why a d-MC derived from one MC can be generated from another MC once again.

1.4.3. A new efficient algorithm for solving d-MCs

Grounded on the obtained results, a new efficient algorithm is suggested to solve all d-MCs. Both complexity analysis and numerical examples are provided to show the efficiency of the proposed algorithm. As demonstrated through theoretical and computational results, the proposed algorithm outperforms the existing algorithms in solving all d-MCs. What is more, a practical case study of the LCD monitor delivery is presented to illustrate the application of the proposed algorithm.

The rest of this paper is organized as follows: Section 2 introduces the network model, the Sum of Disjoint Product method for evaluating R_{d+1}. In Section 3, a new technique is developed to determine lower capacity bounds of arcs based on which an improved mathematical model with respect to d-MCs is built; also, two judging criteria are presented to correctly and effectively detect duplicate d-MCs after analyzing the drawbacks of the existing methods. Grounded on these newfound results, an algorithm for solving d-MCs without duplicates is suggested, together with a discussion on its time complexity. In Section 4, a simple example is adopted to illustrate how the suggested algorithm works, and computational experiments are performed to investigate the performance of the suggested algorithm, together with comparisons with the existing methods. As evidence of the utility of the proposed algorithm, a practical case study regarding LCD monitors delivery is provided in Section 5. The final Section presents some concluding remarks, and discusses the future research.

2. Preliminaries

2.1. The stochastic-flow network model

A stochastic-flow network $G(V, E, W)$ consists of a set of nodes $V = \{1, 2, \ldots, n\}$ with n denoting the number of nodes, a set of arcs $E = \{e_1, e_2, \ldots, e_m\}$ with m denoting the number of arcs, and a largest capacity vector $W = (W_1, W_2, \ldots, W_m)$ with $W_i = W(e_i)$ denoting the max-capacity of e_i for $1 \le i \le m$. The source node and the destination node in $G(V, E, W)$ are represented by 1 and n, respectively. The capacity of e_i is denoted by $X(e_i)$ which takes random integer values from 0 to W_i. A network capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$ indicates the current capacity of all arcs. The max-flow of the network under X (or the network capacity under X) is denoted by $M(X)$, and $M(X)$ is always called the structure function of a stochastic-flow network (Satitsatian and Kapur, 2006; Niu and Xu, 2012; Niu et al., 2014). The max-flow of the network under the largest capacity vector W is denoted by D, i.e., $D = M(W)$, then the following relation is observed: $M(X) \le D$ for any capacity vector X. For example, the network in Fig. 1 shows $V = \{1, 2, 3, 4\}$ with $n = 4$, whereby 1 is the source node and 4 is the destination node, $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ with $m = 6$, and the largest capacity vector $W = (4, 3, 4, 1, 3, 3)$. Considering the largest capacity vector $W = (4, 3, 4, 1, 3, 3)$, the max-flow of the network under W is $M(W) = 10$, thereby $M(X) \le 10$ for any capacity vector X. The notations used throughout this paper are presented in Appendix A.

As with most of the existing literature (Jane et al., 1993; Yeh, 2002, 2008; Yan and Qian, 2007; Yeh et al., 2015), the current study assumes that the network model satisfies the following assumptions: (1) Each node is perfectly reliable; (2) The capacity of each arc e_i $(1 \le i \le m)$ is a non-negative integer-valued random variable which takes values from 0 to W_i according to a given probability distribution; (3) The capacities of different arcs are stochastically independent; (4) All flows in the network obey the conservation law, i.e., total flows into and from a node (not source and destination nodes) are all equal.

Since an unreliable node can be replaced by two reliable nodes and one unreliable arc (refer to Aggarwa et al. (1975), and Jane and Laih (2010) for the replacement), only the network with reliable nodes is discussed here. In addition, note that there exists only one d-MC (i.e., the largest capacity vector W) if $d = D$, thus we merely consider $d < D$.

2.2. Evaluating R_{d+1} in terms of d-MCs

The performance index R_{d+1} is defined as the probability that $d + 1$ units of flow demand can be successfully distributed from the source to the destination. That is, $R_{d+1} = \Pr\{X|M(X) \ge d + 1\} = 1 - \Pr\{X|M(X) \le d\}$. As stated previously, the SDP

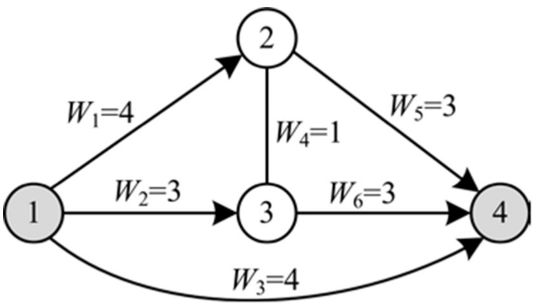

Fig. 1. A stochastic-flow network (Yeh, 2008).

method is available to evaluate R_{d+1} if all d-MCs are determined. Assume $X^1, X^2, \ldots, X^q$ are all d-MCs, and let $A_1 = \{X|X \le X^1\}$, $A_2 = \{X|X \le X^2\}, \ldots, A_q = \{X|X \le X^q\}$, i.e. $A_i = \{X|X \le X^i\}$ $(1 \le i \le q)$ is a set of the state vectors that are smaller than or equal to X^i, then R_{d+1} can be evaluated via the SDP method as follows:

$$\begin{aligned} R_{d+1} &= 1 - \Pr\{X|M(X) \leqslant d\} \\ &= 1 - \Pr\left(A_1 \cup A_2 \cup \ldots \cup A_q\right) \\ &= 1 - \Pr\left(A_1 \cup (A_2 - A_1) \cup \ldots \cup \left(A_q - \cup_{j=1}^{q-1} A_j\right)\right) \\ &= 1 - \sum_{i=1}^{q} \Pr(B_i) \end{aligned} \quad (1)$$

where $B_1 = A_1$, $B_i = A_i - \cup_{j=1}^{i-1} A_j$, $i = 2, 3, \ldots, q$, $\Pr(B_i) = \sum_{X \in B_i} \Pr(X)$, and $\Pr(X) = \prod_{k=1}^{m} \Pr(X(e_k))$.

Recently, some works have been reported to improve the traditional SDP method, but they are beyond the focus of this paper, and readers can refer to Zuo et al. (2007), Bai et al. (2015), and Yeh (2015) for details.

3. The suggested algorithm

A capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC if and only if $M(X) = d$, and $M(X + 0(e_i)) > d$ for each $e_i \in U(X)$ where $0(e_i) = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$, i.e. capacity is 1 for e_i and zero for others, and $U(X) = \{e_i|X(e_i) < W(e_i)\}$ is the set of unsaturated arcs in X. That is, two conditions must be satisfied for a d-MC X: (1) the network capacity under X is d; (2) network capacity is sensitive to the capacity increase of any unsaturated arc, i.e. the increase in capacity of one unsaturated arc results in a larger network capacity (above d). To facilitate understanding the concept of d-MC, we use two examples to illustrate it. Given the demand level $d = 8$, we consider a capacity vector $X = (2, 2, 4, 1, 3, 3)$ of the network in Fig 1. The network under X is shown in Fig. 2 (1), and the network capacity under X is $M(X) = 8$ (refer to Fig. 2 (1)-maxflow). Thus, X satisfies the first condition. It is clear that e_1 and e_2 are unsaturated arcs denoted by dotted lines in Fig. 2 (1). The network under $X + 0(e_1)$ is shown in Fig. 2 (2), and the network capacity under $X + 0(e_1)$ is $M(X + 0(e_1)) = 9$ (refer to Fig. 2 (2)-maxflow). Similarly, the network capacity under $X + 0(e_2)$ is $M(X + 0(e_2)) = 9$ (refer to Fig. 2 (3)-maxflow). Thus, X satisfies the second condition. Because $X = (2, 2, 4, 1, 3, 3)$ satisfies both conditions, it is an 8-MC. We consider another capacity vector $X = (3, 3, 2, 1, 3, 3)$, and Fig. 3 illustrates the network capacity under different cases. It can be seen from Fig. 3 (2)-maxflow that the network capacity under $X + 0(e_1)$ is $M(X + 0(e_1)) = 8 = d$, thereby $X = (3, 3, 2, 1, 3, 3)$ does not satisfies the second condition, i.e. $X = (3, 3, 2, 1, 3, 3)$ is not an 8-MC. Since a d-MC is also a d-MC candidate, the existing methods need to search for all d-MC candidates prior to determining d-MCs. When it is assumed that all MCs are known in advance, the existing algorithms employ the following model proposed by Jane et al. (1993) to search for d-MC candidates.

Lemma 1. *If a capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC, then there exists at least one MC C such that the following conditions are satisfied*:

$$\sum_{e_i \in C} X(e_i) = d \quad (2)$$

$$0 \leqslant X(e_i) \leqslant \mathrm{Min}\{W_i, d\} \text{ for all } e_i \in C \quad (3)$$

$$X(e_i) = W_i \text{ for all } e_i \notin C \quad (4)$$

Each feasible solution to conditions (2)(4) is a d-MC candidate (Jane et al., 1993). By Lemma 1, a d-MC candidate is generated from at least one MC. A d-MC candidate is not necessarily a d-MC, thus there is also a need to verify it. The well-known method for verifying d-MC candidates is based on Lemma 2 (*refer to* Appendix A). In addition, different MCs may generate identical d-MCs, i.e. duplicate d-MCs, so a step to detect and remove duplicate d-MCs is indispensable. In the following subsections, we will detail the vital theoretical results based on which a new efficient algorithm is suggested to solve all d-MCs.

3.1. A novel technique for finding lower capacity bounds of arcs

As mentioned previously, a cost-effective scheme of increasing the efficiency of solving d-MCs is to reduce the number of d-MC candidates. Lemma 1 shows that the number of d-MC candidates is primarily determined by condition (3) which specifies the capacity range of e_i in solving d-MC candidates. Hence, if the capacity range in condition (3) can be narrowed, the number of d-MC candidates will be potentially decreased. To arrive at this aim, some works determine the minimal capacity of e_i, instead of 0, in condition (3) by introducing the concept of lower capacity bound. Lower capacity bound $L(e_i)$ of e_i $(1 \le i \le m)$ is defined as the minimal capacity of e_i, such that the max-flow from the source node to the destination node is equal to d (Yeh, 2008). As a result, $L(e_i)$ can be regarded as a tighter restriction for the capacity bound of e_i in seeking d-MC candidates.

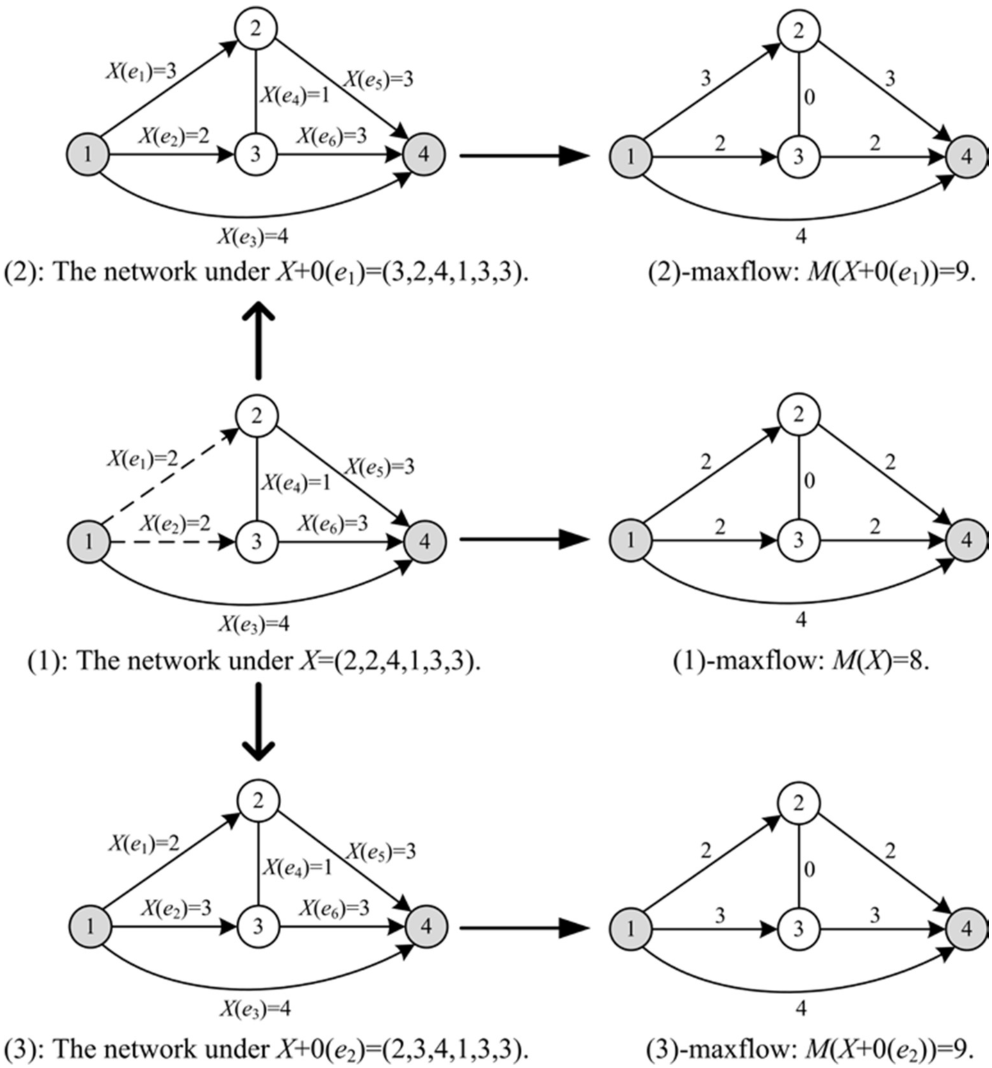

Fig. 2. An illustration of 8-MC $X = (2, 2, 4, 1, 3, 3)$.

Yan and Qian (2007) first proposed a method to find $L(e_i)$ $(1 \le i \le m)$. The time complexity of the method by Yan and Qian is $O(mp\log_2 p)$, where p is the number of MCs. Since the number of MCs can be as large as 2^{n-2} (Shier, 1991), the method of Yan and Qian with the time complexity of $O(mp\log_2 p) = O(mn2^{n-2})$ is inefficient. Yeh (2008) proposed a method to seek $L(e_i)$ $(1 \le i \le m)$ on the basis of the classical binary-search method and max-flow method. The method of Yeh finds $L(e_i)$ $(1 \le i \le m)$ by implementing the max-flow algorithm multiple times. Yeh (2008) demonstrated that his algorithm is more efficient than the one by Yan and Qian (2007). But, there is a minor defect in Yeh's method, such that it may work improperly (Forghani-elahabad and Mahdavi-Amiri, 2013).

According to the definition of lower capacity bound, $L(e_i)$ $(1 \le i \le m)$ actually represents the minimal capacity level the arc e_i should provide to exactly satisfy the flow demand d. For example, Fig. 4 shows $L(e_3) = 2$ when $d = 8$, and if $X(e_3) < L(e_3) = 2$, $M(X) < 8$ for any capacity vector X. As can be seen below, a novel technique is developed to find $L(e_i)$ $(1 \le i \le m)$ by defining a special capacity vector. The proposed technique is based merely on the max-flow algorithm, and the max-flow algorithm is implemented only once to find $L(e_i)$. Thus, the proposed technique is more desirable in the determination of $L(e_i)$ $(1 \le i \le m)$.

Theorem 1. *Given the demand level d $(0 \le d < D)$, let $W(0_i)$ denote a special capacity vector in which capacity level is 0 for e_i $(1 \le i \le m)$ and the largest for other arcs, i.e. $W(0_i) = (W_1, W_2, \ldots, W_{i-1}, 0, W_{i+1}, \ldots, W_m)$, then*

$$L(e_i) = \begin{cases} \text{does not exist} & \text{if } M(W(0_i)) > d \\ 0 & \text{if } M(W(0_i)) = d \\ d - M(W(0_i)) & \text{if } M(W(0_i)) < d \end{cases} \tag{5}$$

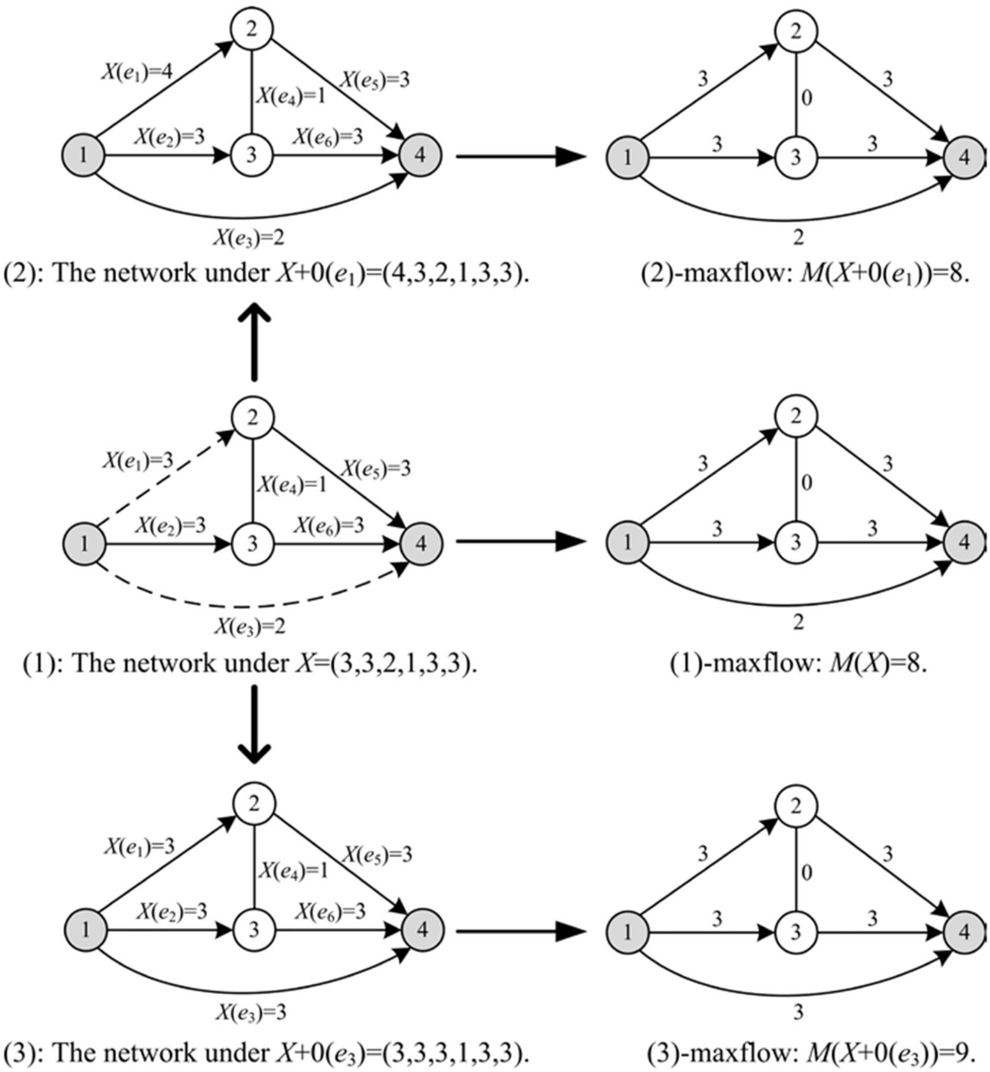

Fig. 3. An illustration of non-8-MC $X = (3, 3, 2, 1, 3, 3)$.

Proof. Obviously, $L(e_i) \geq 0$ holds. If $M(W(0_i)) > d$, it means that even when the capacity of e_i is 0, the max-flow from the source node to the destination node is above d. Thus, there does not exist any $L(e_i)$ satisfying the definition of lower capacity bound, i.e. $L(e_i)$ does not exist.

First, note that given a capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$, if $M(X) = d$, $L(e_i) \leq X(e_i)$ follows from the definition of $L(e_i)$. If $M(W(0_i)) = d$, let $X = W(0_i) = (W_1, W_2, \ldots, W_{i-1}, 0, W_{i+1}, \ldots, W_m)$, then $M(X) = d$. As a result, $L(e_i) \leq X(e_i) = 0$ holds. But since $L(e_i) \geq 0$ holds, $L(e_i) = 0$ can be obtained.

If $M(W(0_i)) < d$, since $M(W) = D > d$, at least $d - M(W(0_i))$ units of flow must travel through arc e_i such that d units of flow can be transmitted from the source node to the destination node. Hence, $L(e_i) = d - M(W(0_i))$.□

For ease of understanding Theorem 1, an example of finding $L(e_3)$ of e_3 in Fig. 1 is presented in Fig. 5. By Theorem 1, it is only necessary to compute $M(U(0_i))$ for finding $L(e_i)$ $(1 \leq i \leq m)$, thereby the time complexity of finding lower capacity bounds of all arcs is $O(mn^2\log^3 n)$, where $O(n^2\log^3 n)$ is the time complexity of calculating the max-flow (Ahuja et al., 1997). Given that $O(mn^2\log^3 n) \ll O(mn2^{n-2})$, the proposed technique derived from Theorem 1 is more efficient than the method by Yan and Qian (2007).

In Theorem 1, if $M(W(0_i)) \leq d$, the value of $L(e_i)$ can be determined; but if $M(W(0_i)) > d$, no value of $L(e_i)$ satisfies the definition of lower capacity bound. In such cases, the lower capacity bound is non-existent. That is, $L(e_i)$ does not always exist. The nonexistence of $L(e_i)$ implies that even if the minimal capacity of e_i is 0, the max-flow from the source node to the destination node is larger than d. The following theorem, compared with Theorem 1, is much more simple in determining the nonexistence of $L(e_i)$ $(1 \leq i \leq m)$.

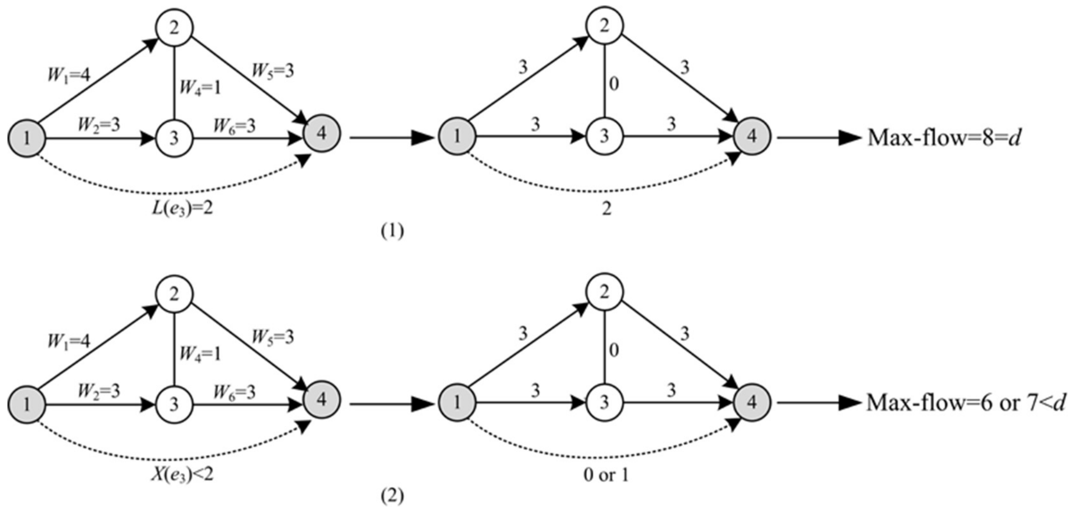

Fig. 4. Lower capacity bound of e_3 when $d = 8$.

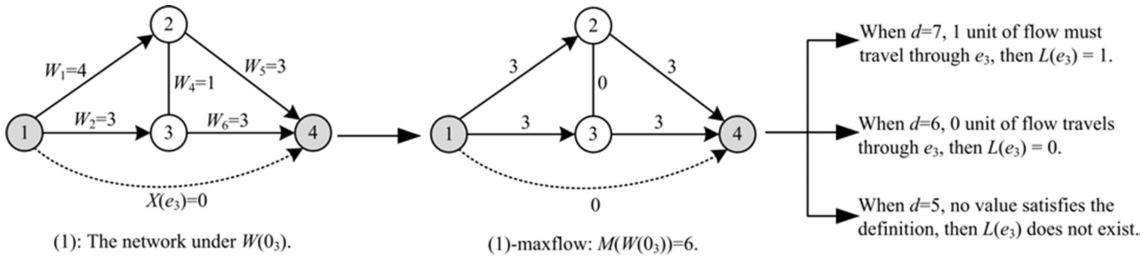

Fig. 5. An illustration of Theorem 1.

Theorem 2. *If $D - W_i > d$, $L(e_i)$ does not exist $(1 \le i \le m)$.*

Proof. By the definition of D, $M(W(0_i))$ and W_i, we have $D - M(W(0_i)) \le W_i$. If $D - W_i > d$, it is easy to obtain $M(W(0_i)) \ge D - W_i > d$, i.e. $M(W(0_i)) > d$. Thus, $L(e_i)$ does not exist $(1 \le i \le m)$ by Theorem 1. □

It should be noted that if $D - W_i \le d$, the existence of $L(e_i)$ $(1 \le i \le m)$ is unspecified. In such a case, Theorem 1 is still required to determine whether $L(e_i)$ $(1 \le i \le m)$ exists or not. Thus, it is more reasonable to combine Theorem 1 and Theorem 2 to find $L(e_i)$ $(1 \le i \le m)$. The role of the lower capacity bound depends on the fact that the minimal capacity of e_i for all $e_i \in C$ in a d-MC derived from C should not be below its lower capacity bound (refer to Corollary 2 in Appendix A). Consequently, the minimal capacity 0 in condition (3) can be replaced by $L(e_i)$ when $L(e_i)$ exists. If $L(e_i)$ does not exist, the minimal capacity 0 of e_i in condition (3) remains the same. Thus, we can obtain an improved model with respect to d-MCs.

Theorem 3. *If a capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC, then there exists at least one MC C such that the following conditions are satisfied:*

$$\sum_{e_i \in C} X(e_i) = d \tag{6}$$

$$L(e_i) \leqslant X(e_i) \leqslant \mathrm{Min}\{W_i, d\} \text{ when } L(e_i) \text{ exists for all } e_i \in C \tag{7}$$

$$0 \leqslant X(e_i) \leqslant \mathrm{Min}\{W_i, d\} \text{ when } L(e_i) \text{ does not exist for all } e_i \in C \tag{8}$$

$$X(e_i) = W_i \text{ for all } e_i \notin C \tag{9}$$

Proof. Directly from Lemma 1 and Corollary 2. □

The goal of introducing lower capacity bounds, other than to cut down the number of d-MC candidates, is also to find some special d-MCs without any tedious verification. Corollary 3 in Appendix A presents these special d-MCs with the distinct feature that there is only one unsaturated component in every d-MC. Since the d-MCs with only one unsaturated component can be determined by Corollary 3, they should be removed from Theorem 3. Thus, the following theorem is at hand.

Theorem 4. *If a capacity vector* $X = (X(e_1), X(e_2), \ldots, X(e_m))$ *is a d-MC with* $|U(X)| > 1$, *then there exists at least one MC C such that the following conditions are satisfied*:

$$\sum_{e_i \in C} X(e_i) = d \tag{10}$$

$$L(e_i) + 1 \leqslant X(e_i) \leqslant \mathrm{Min}\{W_i, d\} \text{ when } L(e_i) \text{ exists for all } e_i \in C \tag{11}$$

$$0 \leqslant X(e_i) \leqslant \mathrm{Min}\{W_i, d\} \text{ when } L(e_i) \text{ does not exist for all } e_i \in C \tag{12}$$

$$X(e_i) = W_i \text{ for all } e_i \notin C \tag{13}$$

Proof. Directly follows from Theorem 3 and Corollary 3. □

As soon as lower capacity bounds of arcs are found, the d-MCs with only one unsaturated component can be directly derived from Corollary 3. Accordingly, the d-MCs with more than one unsaturated component can be solved by Theorem 4. After the determination of d-MCs, the next step is to detect whether they are duplicates.

3.2. Two judging criteria for detecting duplicate d-MCs

The d-MCs are duplicate in the sense that they are obtained multiple times from different MCs. The well-known comparison method is inefficient because it detects a duplicate d-MC by comparing it with all of the other d-MCs (Yeh, 2008; Yeh et al., 2015). Yeh (2008) found that unsaturated components in d-MCs are the key components for detecting duplicates, and proposed a method to detect duplicate d-MCs. Nevertheless, Yeh's method may work improperly in some special cases (refer to the example in Appendix A). Hence, there is a demand for developing new efficient approaches to identify duplicate d-MCs.

To develop an efficient method for identifying duplicate d-MCs, the key is to discover the reason for the generation of duplicate d-MCs, which is precisely neglected by the comparison method. Particularly, we believe that there should exist a specific relationship between the MCs, such that they can generate identical d-MCs. With this in mind, we attempt to explore the underlying reason for duplicates, and present two new judging criteria to identify duplicate d-MCs.

Before providing the judging criteria, we utilize a simple example to show the relationship between two MCs when they generate the same d-MCs. The network in Fig. 1 has 4 MCs, and the related information is shown in Fig. 6 (1). Given the demand level $d = 8$, Fig. 6 (2) describes an 8-MC $X = (\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)$ generated from C_1, and e_2, e_3 are unsaturated arcs whose capacities are underlined. If $X = (\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)$ is also an 8-MC generated from another MC C_j ($j \neq 1$), then C_j must contain e_2 and e_3 (Otherwise, e_2 or e_3 are saturated arcs in all of the 8-MCs generated from C_j according to Eq. (4)). That is, $U(X) = \{e_2, e_3\} \subseteq C_j$ holds. The relationship $\{e_2, e_3\} \subseteq C_j$ means C_3 is the MC that also generates the 8-MC $X = (\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)$. Fig. 6 (3) describes the 8-MC $X = (4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$ generated from C_3. In addition, Fig. 6 (4) demonstrates the following relationship between C_1 and C_3: $\mathrm{Cap}(C_1) = \mathrm{Cap}(C_3) = 11$, i.e., the capacities of C_1 and C_3 are equal. In the following, we formally present two theorems that pinpoint the sufficient and necessary conditions for yielding duplicate d-MCs, and are vitally important to the suggested algorithm.

Theorem 5. *Let* C_i *and* C_j *be two distinct MCs and X be a d-MC generated from* C_i, *X is also a d-MC generated from* C_j, *i.e. X is a duplicate d-MC, if and only if* $Cap(C_i) = Cap(C_j)$, *i.e.* $\sum_{e \in C_i} W(e) = \sum_{e \in C_j} W(e)$, *and* $U(X) \subseteq C_j$.

Proof.

(1) If X is also a d-MC generated from MC C_j, by Corollary 4, it is easy to have

$$\sum_{e \in E} X(e) = d + \sum_{e \in E} W(e) - \sum_{e \in C_i} W(e) = d + \sum_{e \in E} W(e) - \sum_{e \in C_j} W(e).$$

Then, one can obtain

$$\sum_{e \in C_i} W(e) = \sum_{e \in C_j} W(e). \tag{14}$$

Meanwhile, it is clear to have $U(X) \subseteq C_j$ by Corollary 5.

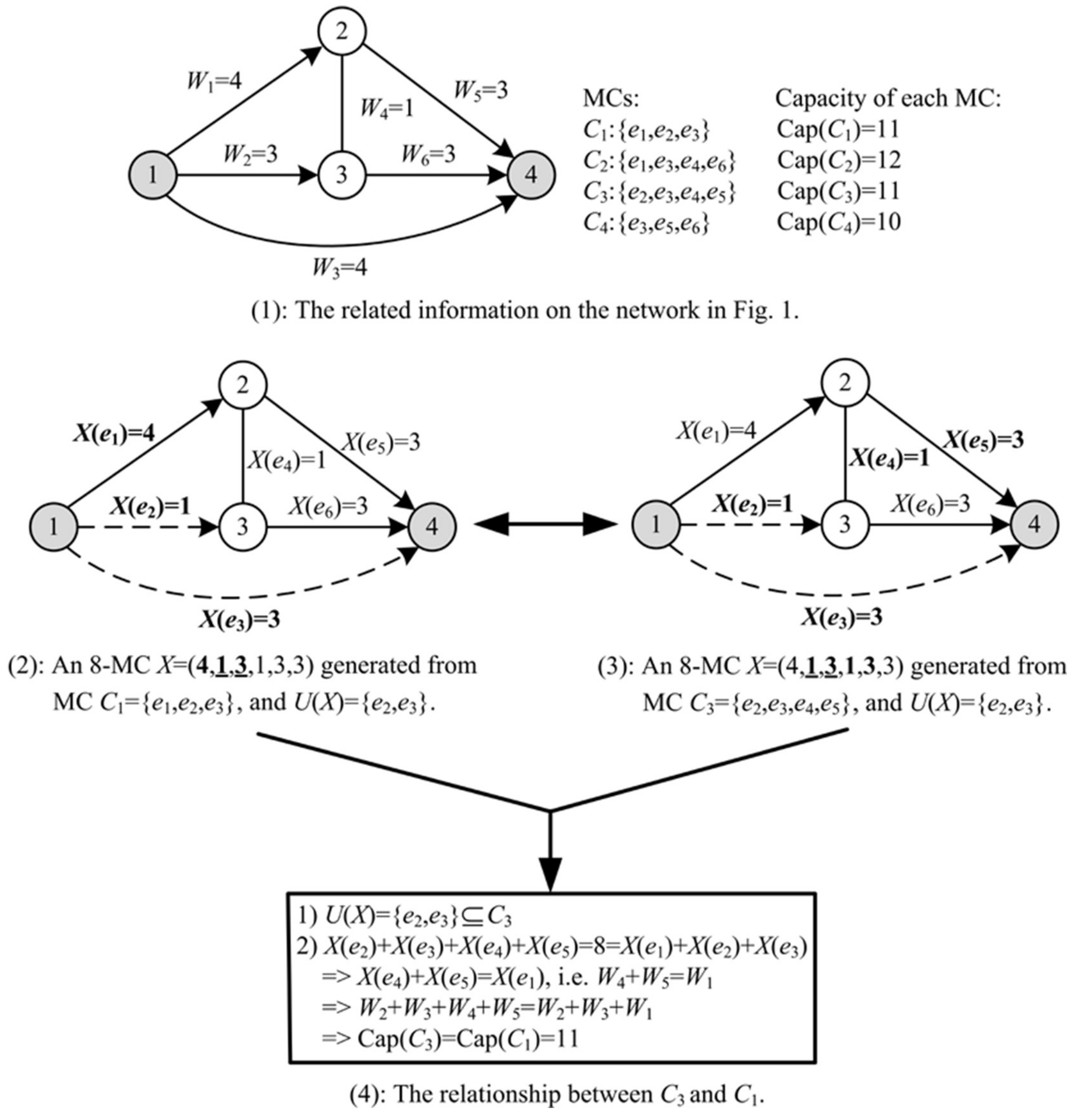

Fig. 6. The relationship between two MCs when generating identical d-MCs.

(2) Suppose $\sum_{e \in C_i} W(e) = \sum_{e \in C_j} W(e)$ and $U(X) \subseteq C_j$.

Since X is a d-MC generated from MC C_i, by Lemma 1, one have

$$d = \sum_{e \in C_i} X(e) = \sum_{e \in U(X)} X(e) + \sum_{e \in (C_i - U(X))} W(e). \tag{15}$$

Now compute $\sum_{e \in C_j} X(e)$, and

$$\sum_{e \in C_j} X(e) = \sum_{e \in (C_j \cap U(X))} X(e) + \sum_{e \in (C_j - U(X))} W(e). \tag{16}$$

Because $U(X) \subseteq C_j$, then

$$\sum_{e \in C_j} X(e) = \sum_{e \in U(X)} X(e) + \sum_{e \in C_j} W(e) - \sum_{e \in U(X)} W(e). \tag{17}$$

Also, because $\sum_{e \in C_i} W(e) = \sum_{e \in C_j} W(e)$, then

$$\sum_{e \in C_j} X(e) = \sum_{e \in U(X)} X(e) + \sum_{e \in C_i} W(e) - \sum_{e \in U(X)} W(e). \tag{18}$$

Meanwhile, because $U(X) \subseteq C_i$, then

$$\sum_{e\in C_j} X(e) = \sum_{e\in U(X)} X(e) + \sum_{e\in(C_i-U(X))} W(e) \tag{19}$$

Based on Eq. (15), one can obtain

$$\sum_{e\in C_j} X(e) = \sum_{e\in c_i} X(e) = d. \tag{20}$$

In addition, $U(X) \subseteq C_j$ means $X(e) = W(e)$ for all $e \notin C_j$. Thus, by Lemma 1, X is a d-MC candidate generated from C_j. As a result, X is also a d-MC derived from MC C_j because X is a d-MC. □

Theorem 5 reveals that for a d-MC X derived from C_i, both the capacity of C_i and the unsaturated components in X are the key points for detecting whether X is a duplicate d-MC. If the capacity of C_i is equal to that of C_j and all of the unsaturated components in X belong to C_j, X can be generated from C_j once again, i.e. X is a duplicate d-MC. Notably, the condition Cap(C_i) = Cap(C_j) is indispensable in detecting duplicate d-MCs. If Cap(C_i) ≠ Cap(C_j), it is impossible for C_i and C_j to generate identical d-MCs. Thus, there is no need to check whether the d-MCs generated from C_i (C_j) are duplicate d-MCs derived from C_j (C_i) when Cap(C_i) ≠ Cap(C_j). Besides, given a d-MC X, since unsaturated components and saturated components in X are complementary (i.e. every component in X is either unsaturated component or saturated component), we can try to detect duplicate d-MCs in terms of saturated components. For example, it can be seen from Fig. 6 that $(C_1 - C_3) = \{e_1\}$ and $X(e_1) = W(e_1)$, i.e. all of the arcs belonging to $(C_1 - C_3)$ are saturated when X is also an 8-MC generated from C_3. Consequently, we present the other theorem.

Theorem 6. *Let C_i and C_j be two distinct MCs and X be a d-MC generated from C_i, X is also a d-MC generated from C_j, i.e. X is a duplicate d-MC, if and only if Cap(C_i) = Cap(C_j), i.e. $\sum_{e\in C_i} W(e) = \sum_{e\in C_j} W(e)$, and $X(e) = W(e)$ for all $e \in (C_i - C_j)$.*

Proof. Here, it is only necessary to prove $U(X) \subseteq C_j$ is equivalent to $X(e) = W(e)$ for all $e \in (C_i - C_j)$.

(1) Since X is a d-MC generated from C_i, $U(X) \subseteq C_i$ follows from Corollary 5. Thus, if $U(X) \subseteq C_j$, we can obtain $U(X) \subseteq (C_i \cap C_j)$, which implies $X(e) = W(e)$ for all $e \in (C_i - C_j)$.
(2) That $X(e) = W(e)$ for all $e \in (C_i - C_j)$ means $U(X) \subseteq (C_i \cap C_j)$. Also, it is trivial to have $(C_i \cap C_j) \subseteq C_j$. Hence, $U(X) \subseteq C_j$ holds. □

It is noteworthy that the method of Yeh et al. (2015) detects duplicate d-MCs using only the condition "$X(e) = W(e)$ for all $e \in (C_i - C_j)$", so the other condition Cap(C_i) = Cap(C_j) is also neglected by Yeh et al. (2015). As a result, both methods by Yeh (2008) and Yeh et al. (2015) fail to correctly detect duplicate d-MCs. Now Theorems 5 and 6 have provided two judging criteria for detecting duplicate d-MCs, and both theorems reveal that just because there exists a special relationship between two MCs, they can generate identical d-MCs. Therefore, the two theorems not only provide two approaches to correctly detect duplicate d-MCs, but also explicitly answer the question of why a d-MC can be generated from distinct MCs. In this sense, Theorems 5 and 6 provide new insights into the reason for the generation of duplicate d-MCs, which is always ignored by the comparison-based methods (Lin, 2002; Yan and Qian, 2007; Forghani-elahabad and Mahdavi-Amiri, 2014) which detect a duplicate d-MC by merely comparing it with all of the other d-MCs.

3.3. A new algorithm for solving all d-MCs without duplicates

Grounded on the above discussions, an algorithm for solving all d-MCs without duplicates is provided as follows.

input: All MCs $C_1, C_2, \ldots, C_p$ in a stochastic-flow network $G(V, E, W)$ and demand level d.
output: All d-MCs without duplicates.
Step 0. Calculate Cap(C_i) for all $i = 1, 2, \ldots, p$, and let $D = \min\{\text{Cap}(C_i)|i = 1, 2, \ldots, p\}$.
Step 1. If $D - W_i > d$, $L(e_i)$ does not exist, otherwise, find $L(e_i)$ by Theorem 1, where $1 \le i \le m$. $\Omega = \{X|X = (W_1, W_2, \ldots, W_{i-1}, L(e_i), W_{i+1}, \ldots, W_m)$ if $L(e_i)$ exists for $1 \le i \le m\}$.
Step 2. All MCs are grouped by their capacities, such that the MCs with identical capacity are put into one group. Suppose that all MCs are identified as λ groups: $\Phi_1, \Phi_2, \ldots, \Phi_\lambda$, and the number of MCs in the group Φ_k ($1 \le k \le \lambda$) is p_k (then, $p_1 + p_2 + \cdots + p_\lambda = p$).
Step 3. $k = 1$.
Step 4. Solve all d-MCs from the MCs in Φ_k according to the following steps:
Step 4.1. Solve all d-MCs from the first MC C_{k_1} using Eqs. (10)–(13) and Lemma 2, and let $\Omega = \Omega \cup \{d\text{-MCs from } C_{k_1}\}$. If $p_k = 1$, go to **Step 5**.
Step 4.2. $i = 2$.

Table 1
Time complexities of different algorithms.

Algorithm	Time complexity
Jane et al. (1993), Lin (2002), Yeh (2002, 2004), and Yan and Qian (2007)	$O(mp^2\sigma^2)$
Yeh (2008), Forghani-elahabad and Mahdavi-Amiri (2014), and Yeh et al. (2015)	$O(mp^2\sigma)$
The proposed algorithm	$O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right)$

Note: $p_1 + p_2 + \cdots + p_\lambda = p$, and $\sum_{k=1}^{\lambda} mp_k^2\sigma \leq O(mp^2\sigma) < O(mp^2\sigma^2)$.

Step 4.3. find all d-MC candidates, say X_{k_ij} where $j = 1, 2, \ldots, J_1$, from the ith MC C_{k_i} using Eqs. (10)–(13). If no d-MC candidate exists, go to **Step 4.7**.
Step 4.4. If $\text{Cap}(C_{k_i}) = D$, X_{k_ij} is a d-MC where $j = 1, 2, \ldots, J_1$, and go to **Step 4.6**.
Step 4.5. Use Lemma 2 to check whether X_{k_ij} is a d-MC where $j = 1, 2, \ldots, J_1$. If none of them is a d-MC, go to **Step 4.7**; otherwise, suppose X_{k_ij} is a d-MC where $j = 1, 2, \ldots, J_2$.
Step 4.6. For every d-MC X_{k_ij} where $j = 1, 2, \ldots, J_2$ (J_1), if there exists one r, $1 \leq r \leq i - 1$, such that $U(X_{k_ij}) \subseteq C_{k_r}$ (or $X_{k_ij}(e) = W(e)$ for all $e \in (C_{k_i} - C_{k_r})$), X_{k_ij} is a duplicate d-MC; otherwise, $\Omega = \Omega \cup \{X_{k_ij}\}$.
Step 4.7. If $i < p_k$, let $i = i + 1$, and return to **Step 4.3**.
Step 5. If $k < \lambda$, let $k = k + 1$, and return to **Step 4**; otherwise, Ω is the set of all d-MCs and stop.

Note that Steps 0, 1, and 2 can be regarded as the pre-processing steps, in which the capacity of each MC is calculated (Step 0), lower capacity bounds of arcs and the d-MCs with only one unsaturated component are determined (Step 1), and all MCs are distinguished by their capacities (Step 2). The aim of grouping MCs in Step 2 is to reduce the cost of detecting duplicate d-MCs in the subsequent steps, because there is only need to verify whether the MCs from the same group generate duplicate d-MCs. Step 4 is the key step for solving all d-MCs with more than one unsaturated component. Of note is that for the first MC in each group Φ_k $(1 \leq k \leq \lambda)$, there is no need to detect duplicate d-MCs (Step 4.1). Step 4.3 is to solve all d-MC candidates from the ith MC C_{k_i}. Steps 4.4 and 4.5 are used to check whether the obtained d-MC candidates are d-MCs. The aim of Step 4.6 is to detect and remove duplicate d-MCs. Step 4.7 is to control the iteration for solving MCs in the group Φ_k, and Step 6 is to control the iteration for solving the group Φ_k.

The time complexity of the suggested algorithm is discussed as follows: Step 0 needs $O(mp)$ time to calculate $\text{Cap}(C_i)$ and D, where $i = 1, 2, \ldots, p$. Step 1 requires $O(mn^2\log^3 n)$ time to find all lower capacity bounds, where $O(n^2\log^3 n)$ is the time complexity for computing the max-flow (Ahuja et al., 1997), and $O(m)$ time to find the d-MCs with only one unsaturated component, so Step 1 requires $O(mn^2\log^3 n)$ time. Step 2 takes $O(p\log p)$ time to sort and group all MCs. Steps 3 and 5 take $O(1)$ time. Therefore, the time complexity of Steps 0 to 3 is $O(p(\log p + m))$. Step 4.1 takes $O((m^2 + n^2\log^3 n)\sigma)$ time to derive d-MCs from the first MC C_{k_1} (Yeh, 2008). Steps 4.2, 4.4, and 4.7 require $O(1)$ time. Note that $p_k\sigma$ is the total number of d-MC candidates obtained from Step 4.3, and it takes $O(m^2 + n^2\log^3 n)$ time to verify whether a d-MC candidate is a d-MC in Step 4.5 (Yeh, 2002). As a result, the time complexity of Step 4.5 is $O((m^2 + n^2\log^3 n)(p_k - 1)\sigma)$. For a d-MC X_{k_ij} derived from C_{k_i}, it takes at most $O((i-1)m)$ time to detect whether it is a duplicate in Step 4.6, where $i - 1$ denotes the maximum times for checking whether there exists one r, such that $U(X_{k_ij}) \subseteq C_{k_r}$ (or $X_{k_ij}(e) = W(e)$ for all $e \in (C_{k_i} - C_{k_r})$). Then, Step 4.6 requires $O(m\sigma(1 + 2 + \ldots + p_k - 1)) = O(mp_k^2\sigma)$ amount of time to detect all of the duplicate d-MCs in the group Φ_k in the worst case. Consequently, The time complexity of Step 4 is $O(mp_k^2\sigma)$. Note that there are λ groups, and the total time complexity of Step 3 to Step 5 is thus $O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right)$. Therefore, the time complexity of the suggested algorithm is $O(p(\log p + m)) + O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right) = O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right)$. That is, the time complexity of the suggested algorithm for solving all d-MCs without duplicates is $O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right)$, where m is the number of arcs, p_k is the number of MCs in the group Φ_k and $p_1 + p_2 + \cdots + p_\lambda = p$ where p isthe number of MCs, $\sigma = \text{Min}\left\{\binom{m+d-1}{d}, \prod_{e_i \in E}(\text{Min}\{W(e_i), d\} + 1)\right\}$ is the number of d-MC candidates derived from each MC (Yeh, 2008).

Note that the time complexity of the methods by Jane et al. (1993), Lin (2002), Yeh (2002, 2004), and Yan and Qian (2007) is $O(mp^2\sigma^2)$, and the time complexity of the methods by Yeh (2008), Forghani-elahabad and Mahdavi-Amiri (2014), and Yeh et al. (2015) is $O(mp^2\sigma)$. In view of the fact that $O\left(\sum_{k=1}^{\lambda} mp_k^2\sigma\right) = O(m\sigma(p_1^2 + p_2^2 + \cdots + p_\lambda^2)) \leq O(m\sigma(p_1 + p_2 + \cdots + p_\lambda)^2) = O(mp^2\sigma) < O(mp^2\sigma^2)$, it is trivial to conclude that the suggested algorithm is more efficient than the existing methods. For comparison, the time complexities of different algorithms are presented in Table 1.

4. Numerical examples

4.1. An illustrative example

To elucidate the suggested algorithm, a simple network in Fig. 1 is adopted to trace the steps of the algorithm. Fig. 1 contains 4 nodes and 6 arcs, and is adopted from Yeh (2008). There are 4 MCs in Fig. 1: $C_1 = \{e_1, e_2, e_3\}$, $C_2 = \{e_1, e_3, e_4, e_6\}$, $C_3 = \{e_2,$

$e_3, e_4, e_5\}$, and $C_4 = \{e_3, e_5, e_6\}$. The largest capacities of arcs $e_1, e_2, e_3, e_4, e_5, e_6$ are $W(e_1) = 4$, $W(e_2) = 3$, $W(e_3) = 4$, $W(e_4) = 1$, $W(e_5) = 3$, and $W(e_6) = 3$, respectively. The capacity probabilities of all arcs in Fig. 1 are presented in Table 2. Provided that the demand level is 9, the reliability index R_9 can be evaluated in terms of 8-MCs (i.e. $d = 8$). The following procedure describes how to obtain all 8-MCs. After finding all 8-MCs, R_9 are calculated using the SDP method. To facilitate the understanding of the whole procedure, two special notations are used throughout the step-by-step solutions:

(1) "$\boldsymbol{X(e_i)}$" denotes that e_i is in the related MC and $X(\boldsymbol{e_i}) = W(e_i)$.
(2) "$\underline{\boldsymbol{X(e_i)}}$" denotes that e_i is in the related MC and $\underline{X(\boldsymbol{e_i})} < W(e_i)$.

Solve:

Step 0. $\text{Cap}(C_1) = 11$, $\text{Cap}(C_2) = 12$, $\text{Cap}(C_3) = 11$, $\text{Cap}(C_4) = 10$, and $D = 10$.
Step 1. $10 - W_i < 8$ for $i = 1, 2, 3, 5, 6$, then, according to Theorem 1, it is easy to obtain $L(e_1) = 1$, $L(e_2) = 0$, $L(e_3) = 2$, $L(e_5) = 1$, and $L(e_6) = 1$. $10 - W_4 = 9 > 8$, thus $L(e_4)$ does not exist. As a result, $\Omega = \{(\underline{\mathbf{1}}, 3, 4, 1, 3, 3), (4, \underline{\mathbf{0}}, 4, 1, 3, 3), (4, 3, \underline{\mathbf{2}}, 1, 3, 3), (4, 3, 4, 1, \underline{\mathbf{1}}, 3), (4, 3, 4, 1, 3, \underline{\mathbf{1}})\}$. The value of $L(e_i)$ $(1 \le i \le 6)$ and the corresponding 8-MCs are shown in Table 3.
Step 2. 4 MCs are grouped as follows: $\Phi_1 = \{C_4\}$, and $p_1 = 1$; $\Phi_2 = \{C_1, C_3\}$, and $p_2 = 2$, and $\Phi_3 = \{C_2\}$, and $p_3 = 1$; $\lambda = 3$.
Step 3. $k = 1$.
Step 4. Solve all 8-MCs from the MCs in Φ_1 according to the following steps:
Step 4.1. As C_4 is the first MC in Φ_1, solve all 8-MCs from C_4 using Eqs. (10)–(13) and Lemma 2, and obtain three 8-MCs: $(4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$, $(4, 3, \underline{\mathbf{3}}, 1, \mathbf{3}, \underline{\mathbf{2}})$, and $(4, 3, \mathbf{4}, 1, \underline{\mathbf{2}}, \underline{\mathbf{2}})$. Then let $\Omega = \Omega \cup \{(4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3}), (4, 3, \underline{\mathbf{3}}, 1, \mathbf{3}, \underline{\mathbf{2}}), (4, 3, \mathbf{4}, 1, \underline{\mathbf{2}}, \underline{\mathbf{2}})\}$. $p_1 = 1$, then go to **Step 5**.
Step 5. $k = 1 < \lambda$, let $k = 2$, and return to **Step 4**.
Step 4. Solve all 8-MCs from the MCs in Φ_2 according to the following steps:
Step 4.1. As C_1 is the first MC in Φ_2, solve all 8-MCs from C_1 using Eqs. (10)–(13) and Lemma 2, and obtain five 8-MCs: $(\underline{\mathbf{2}}, \underline{\mathbf{2}}, \mathbf{4}, 1, 3, 3)$, $(\underline{\mathbf{2}}, \mathbf{3}, \underline{\mathbf{3}}, 1, 3, 3)$, $(\underline{\mathbf{3}}, \underline{\mathbf{1}}, \mathbf{4}, 1, 3, 3)$, $(\underline{\mathbf{3}}, \underline{\mathbf{2}}, \underline{\mathbf{3}}, 1, 3, 3)$, and $(\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)$. Then let $\Omega = \Omega \cup \{(\underline{\mathbf{2}}, \underline{\mathbf{2}}, \mathbf{4}, 1, 3, 3), (\underline{\mathbf{2}}, \mathbf{3}, \underline{\mathbf{3}}, 1, 3, 3), (\underline{\mathbf{3}}, \underline{\mathbf{1}}, \mathbf{4}, 1, 3, 3), (\underline{\mathbf{3}}, \underline{\mathbf{2}}, \underline{\mathbf{3}}, 1, 3, 3), (\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)\}$. $p_2 \neq 1$.
Step 4.2. $i = 2$.
Step 4.3. C_3 is the second MC in Φ_2, then find all 8-MC candidates from C_3 using Eqs. (10)–(13), and obtain seven 8-MC candidates: $(4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{1}}, \mathbf{4}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{1}}, \mathbf{4}, \mathbf{1}, \underline{\mathbf{2}}, 3)$, $(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \mathbf{1}, \underline{\mathbf{2}}, 3)$, $(4, \underline{\mathbf{2}}, \mathbf{4}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$, $(4, \mathbf{3}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$.
Step 4.4. $\text{Cap}(C_3) = 11 \neq D$.
Step 4.5. Use Lemma 2 to check the obtained seven 8-MC candidates, and obtain $(4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{1}}, \mathbf{4}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{1}}, \mathbf{4}, \mathbf{1}, \underline{\mathbf{2}}, 3)$, $(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, and $(4, \underline{\mathbf{2}}, \mathbf{4}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$ are 8-MCs.
Step 4.6. For the 8-MC $(4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$, when $r = 1$, $U((4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)) = \{e_2, e_3\} \subseteq C_1 = \{e_1, e_2, e_3\}$, then $(4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$ is a duplicate 8-MC. But, $(4, \underline{\mathbf{1}}, \mathbf{4}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, $(4, \underline{\mathbf{1}}, \mathbf{4}, \mathbf{1}, \underline{\mathbf{2}}, 3)$, $(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \mathbf{3}, 3)$, and $(4, \underline{\mathbf{2}}, \mathbf{4}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$ are not duplicate 8-MCs, then let $\Omega = \Omega \cup \{(4, \underline{\mathbf{1}}, \mathbf{4}, \underline{\mathbf{0}}, \mathbf{3}, 3), (4, \underline{\mathbf{1}}, \mathbf{4}, \mathbf{1}, \underline{\mathbf{2}}, 3), (4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \mathbf{3}, 3), (4, \underline{\mathbf{2}}, \mathbf{4}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)\}$.
Step 4.7. $i = 2 = p_2$.
Step 5. $k = 2 < \lambda$, let $k = 3$, and return to **Step 4**.

Table 2
Capacities & capacity probabilities of arcs in Fig. 1.

Arc	Capacity					Capacity probabilities				
e_1	0	1	2	3	4	0.01	0.01	0.03	0.05	0.90
e_2	0	1	2	3	–	0.01	0.02	0.02	0.95	–
e_3	0	1	2	3	4	0.01	0.01	0.03	0.05	0.90
e_4	0	1	–	–	–	0.02	0.98	–	–	–
e_5	0	1	2	3	–	0.01	0.02	0.02	0.95	–
e_6	0	1	2	3	–	0.01	0.02	0.02	0.95	–

Table 3
Lower capacity bounds and the corresponding 8-MCs.

e_i	$L(e_i)$	The corresponding 8-MCs
e_1	1	$(\underline{\mathbf{1}}, 3, 4, 1, 3, 3)$
e_2	0	$(4, \underline{\mathbf{0}}, 4, 1, 3, 3)$
e_3	2	$(4, 3, \underline{\mathbf{2}}, 1, 3, 3)$
e_4	Does not exist	–
e_5	1	$(4, 3, 4, 1, \underline{\mathbf{1}}, 3)$
e_6	1	$(4, 3, 4, 1, 3, \underline{\mathbf{1}})$

Step 4. Solve all 8-MCs from the MCs in Φ_3 according to the following steps:

Step 4.1. As C_2 is the first MC in Φ_3, solve all 8-MCs from C_2 using Eqs. (10)–(13) and Lemma 2, and obtain one 8-MCs: $(\underline{\mathbf{2}}, 3, \mathbf{4}, \underline{\mathbf{0}}, 3, \underline{\mathbf{2}})$. Then let $\Omega = \Omega \cup \{(\underline{\mathbf{2}}, 3, \mathbf{4}, \underline{\mathbf{0}}, 3, \underline{\mathbf{2}})\}$. $p_1 = 1$, then go to **Step 5**.

Step 5. $k = 3 = \lambda$, then Ω is the set of all 8-MCs, and stop.

The final results are indicated in Table 4. Then, according to the obtained 8-MCs and the SDP method, the performance index R_9 can be readily calculated: $R_9 = 0.860262$.

4.2. Computational experiments

In the previous section, we stated that the suggested algorithm is grounded on the newly obtained results, i.e. (1) a new technique for determining lower capacity bounds of arcs; (2) two judging criteria for detecting duplicate d-MCs. The detailed theoretical analyses have demonstrated that the suggested algorithm holds a performance advantage over the existing algorithms. To further explore the performance of the suggested algorithm, this section conducts computational experiments to compare it with several typical algorithms, i.e. Lin's algorithm (2002), Yan and Qian's algorithm (2007), Forghani-elahabad and Mahdavi-Amiri's algorithm (2014), and the algorithm of Yeh et al. (2015). Since one condition for detecting duplicate d-MCs is neglected by Yeh et al. (2015), we add it to their method. All of the algorithms are coded into MATLAB programs, and implemented on a PC with Intel(R) Core (TM) i5-3210 M 2.50 GHz CPU. In addition, the suggested algorithm is identified by which criterion is used to detect duplicates as two types: the suggested algorithm using Theorem 5 to detect duplicates, and the suggested algorithm using Theorem 6 to detect duplicates.

Due to the NP-hard nature of the d-MC problem and the limitations of the PC, we choose one medium-sized network (Fig. 7 (1)) and two relatively larger networks (Fig. 7 (2), (3)) as benchmark networks to conduct the numerical experiments. The three networks are cited from Soh and Rai (1993), and Ramirez-Marquez and Coit (2005b). To make comprehensive comparisons, four different demand levels are solved for each benchmark network, i.e. $d = 5$ (5-MC), $d = 7$ (7-MC), $d = 9$ (9-MC), and $d = 11$ (11-MC). Experimental results on the number of d-MC candidates and the computational time are summarized in Tables 5 and 6. Additionally, to clearly exhibit the performance of different algorithms, the relative ratios of computational times are indicated in Table 7. From Tables 5–7, the following observations are made:

Table 4
The 8-MC candidates and 8-MCs with more than one unsaturated component.

Φ_1: $C_4 = \{e_3, e_5, e_6\}$: $x_3 + x_5 + x_6 = 8$				$x_1 = 4$	8-MC candidate	8-MC?	A duplicate?
$3 \le x_3 \le 4$	$2 \le x_5 \le 3$	$2 \le x_6 \le 3$		$x_2 = 3$ $x_4 = 1$			
3	2	3			$(4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$	Yes	No
3	3	2			$(4, 3, \underline{\mathbf{3}}, 1, \mathbf{3}, \underline{\mathbf{2}})$	Yes	No
4	2	2			$(4, 3, \mathbf{4}, 1, \underline{\mathbf{2}}, \underline{\mathbf{2}})$	Yes	No
Φ_2: $C_1 = \{e_1, e_2, e_3\}$: $x_1 + x_2 + x_3 = 8$				$x_4 = 1$	8-MC candidate	8-MC?	A duplicate?
$2 \le x_1 \le 4$	$1 \le x_2 \le 3$	$3 \le x_3 \le 4$		$x_5 = 3$ $x_6 = 3$			
2	2	4			$(\underline{\mathbf{2}}, \underline{\mathbf{2}}, \mathbf{4}, 1, 3, 3)$	Yes	No
2	3	3			$(\underline{\mathbf{2}}, \mathbf{3}, \underline{\mathbf{3}}, 1, 3, 3)$	Yes	No
3	1	4			$(\underline{\mathbf{3}}, \underline{\mathbf{1}}, \mathbf{4}, 1, 3, 3)$	Yes	No
3	2	3			$(\underline{\mathbf{3}}, \underline{\mathbf{2}}, \underline{\mathbf{3}}, 1, 3, 3)$	Yes	No
4	1	3			$(\mathbf{4}, \underline{\mathbf{1}}, \underline{\mathbf{3}}, 1, 3, 3)$	Yes	No
Φ_2: $C_3 = \{e_2, e_3, e_4, e_5\}$: $x_2 + x_3 + x_4 + x_5 = 8$				$x_1 = 4$	8-MC candidate	8-MC?	A duplicate?
$1 \le x_2 \le 3$	$3 \le x_3 \le 4$	$0 \le x_4 \le 1$	$2 \le x_5 \le 3$	$x_6 = 3$			
1	3	1	3		$(4, \underline{\mathbf{1}}, \underline{\mathbf{3}}, \mathbf{1}, \mathbf{3}, 3)$	Yes	Yes
1	4	0	3		$(4, \underline{\mathbf{1}}, \mathbf{4}, \underline{\mathbf{0}}, \mathbf{3}, 3)$	Yes	No
1	4	1	2		$(4, \underline{\mathbf{1}}, \mathbf{4}, \mathbf{1}, \underline{\mathbf{2}}, 3)$	Yes	No
2	3	0	3		$(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \mathbf{3}, 3)$	Yes	No
2	3	1	2		$(4, \underline{\mathbf{2}}, \underline{\mathbf{3}}, \mathbf{1}, \underline{\mathbf{2}}, 3)$	No	–
2	4	0	2		$(4, \underline{\mathbf{2}}, \mathbf{4}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$	Yes	No
3	3	0	2		$(4, \mathbf{3}, \underline{\mathbf{3}}, \underline{\mathbf{0}}, \underline{\mathbf{2}}, 3)$	No	–
Φ_3: $C_2 = \{e_1, e_3, e_4, e_6\}$: $x_1 + x_3 + x_4 + x_6 = 8$				$x_2 = 3$	8-MC candidate	8-MC?	A duplicate?
$2 \le x_1 \le 4$	$3 \le x_3 \le 4$	$0 \le x_4 \le 1$	$2 \le x_6 \le 3$	$x_5 = 3$			
2	3	0	3		$(\underline{\mathbf{2}}, 3, \underline{\mathbf{3}}, \underline{\mathbf{0}}, 3, \mathbf{3})$	No	–
2	3	1	2		$(\underline{\mathbf{2}}, 3, \underline{\mathbf{3}}, \mathbf{1}, 3, \underline{\mathbf{2}})$	No	–
2	4	0	2		$(\underline{\mathbf{2}}, 3, \mathbf{4}, \underline{\mathbf{0}}, 3, \underline{\mathbf{2}})$	Yes	No
3	3	0	2		$(\underline{\mathbf{3}}, 3, \underline{\mathbf{3}}, \underline{\mathbf{0}}, 3, \underline{\mathbf{2}})$	No	–

(1) The number of d-MC candidates generated by the proposed algorithm is equal to that generated by the algorithms of Yan and Qian (2007), and Forghani-elahabad and Mahdavi-Amiri (2014), but is smaller than or equal to that generated by the algorithms of Lin (2002) and Yeh et al. (2015). In view of the fact that the concept of lower capacity bound is also used by Yan and Qian, and Forghani-elahabad and Mahdavi-Amiri, but is not employed by Lin, and Yeh et al., it can be concluded that the usage of lower capacity bound tends to reduce the number of d-MC candidates. This result well explains why there is a need to develop efficient methods for finding lower capacity bounds of arcs. Moreover, the difference between the algorithms with and without the use of lower capacity bound will become more prominent as the demand d grows.

(2) Note that T_{new1}/T_{new2} represents the relative efficiency of Theorem 5 and theorem 6 in detecting duplicate d-MCs, and thus Theorem 5 appears to be more efficient than Theorem 6, and is more applicable to detecting duplicate d-MCs.

(3) As expected, the suggested algorithm, regardless of which judging criterion (i.e. Theorem 5 or Theorem 6) is used, always outperforms the existing algorithms in solving all d-MCs (d = 5, 7, 9, 11) of the benchmark networks. This result is totally consistent with the theoretical analysis presented in Section 3.3.

(4) The suggested algorithm shows a huge advantage over the algorithms of Lin (2002), Yan and Qian (2007), Forghani-elahabad and Mahdavi-Amiri (2014) which employ the comparison method to detect duplicate d-MCs. To be worthy of attention, while the number of d-MC candidates generated by the suggested algorithm is identical with that generated by Yan and Qian, Forghani-elahabad and Mahdavi-Amiri, the suggested algorithm is significantly more efficient than the methods of Yan and Qian, and Forghani-elahabad and Mahdavi-Amiri. This result is understandable in view of the superiority of the two judging criteria over the comparison method in detecting duplicate d-MCs. To identify a duplicate d-MC, the comparison method needs to inefficiently compare it with all of the other d-MCs whose number is usually enormous. In contrast, the proposed judging criteria really find out the fundamental reason for yielding duplicate d-MCs. Specifically, the judging criteria reveal that only the d-MCs derived from MCs with identical capacity need to be checked, which largely in part enhances the efficiency of identifying duplicates, and that either the unsaturated components or the saturated components in d-MCs can be used to determine duplicates, which is entirely different from the comparison method.

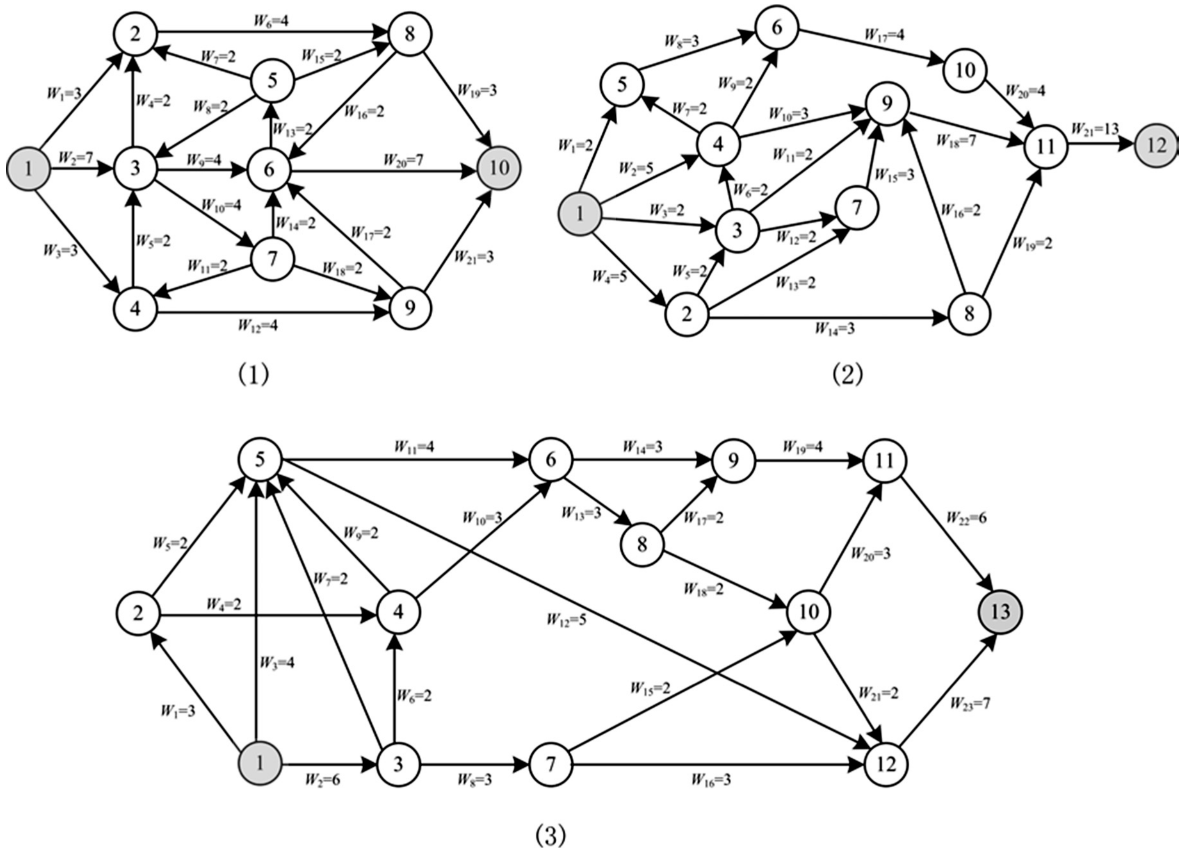

Fig. 7. Benchmark networks (Soh and Rai, 1993; Ramirez-Marquez and Coit, 2005b).

Table 5
The number of *d*-MC candidates generated by different algorithms.

Net ID	*m*	*n*	*p*	*d*	No. of *d*-MC candidates	
					$\sigma_{new} = \sigma_{YQ} = \sigma_{FM}$	$\sigma_L = \sigma_{YBH}$
1	21	10	58	5	8930	8930
				7	10,265	18,915
				9	10,561	27,663
				11	2118	30,684
2	21	12	111	5	27,063	27,063
				7	58,509	59,883
				9	66,144	81,321
				11	27,014	70,644
3	23	13	140	5	39,621	39,621
				7	93,801	107,403
				9	109,578	192,260
				11	18,391	238,944

Note: σ_{new}, σ_L, σ_{YQ}, σ_{FM}, and σ_{YBH} are the number of *d*-MC candidates generated by the proposed algorithm, Lin's algorithm (2002), Yan and Qian's algorithm (2007), Forghani-elahabad and Mahdavi-Amiri's algorithm (2014), and the algorithm of Yeh et al. (2015), respectively.

Table 6
The computational time of different algorithms.

Net ID	*m*	*n*	*p*	*d*-MC	Computational time (in CPU second)					
					T_{new1}	T_{new2}	T_L	T_{YQ}	T_{FM}	T_{YBH}
1	21	10	58	5-MC	4.258	5.345	83.255	83.289	18.604	5.604
				7-MC	4.293	5.714	206.328	85.775	21.495	7.247
				9-MC	3.166	4.106	198.224	46.498	16.299	6.765
				11-MC	0.655	0.868	64.405	1.802	3.335	5.401
2	21	12	111	5-MC	16.753	29.337	813.263	813.285	212.359	31.051
				7-MC	24.949	40.754	2204.765	2054.746	340.908	42.543
				9-MC	19.928	28.021	1815.567	1415.052	189.528	31.332
				11-MC	5.925	6.908	422.149	116.938	43.764	14.349
3	23	13	140	5-MC	28.103	38.882	859.071	859.112	127.815	41.156
				7-MC	36.252	45.761	1792.282	1310.053	203.868	49.341
				9-MC	27.652	29.673	1073.176	525.361	193.595	51.689
				11-MC	3.931	4.165	237.842	12.404	32.188	69.302

Note: T_{new1}, and T_{new2} are the running times of the suggested algorithm using Theorem 8, and Theorem 9, respectively, to detect and remove duplicate *d*-MCs; T_L, T_{YQ}, T_{FM}, and T_{YBH} are the running times of Lin's algorithm (2002), Yan and Qian's algorithm (2007), Forghani-elahabad and Mahdavi-Amiri's algorithm (2014), and the algorithm of Yeh et al. (2015), respectively.

Table 7
The relative performance of different algorithms.

Net ID	*m*	*n*	*p*	*d*-MC	Relative performance				
					T_{new1}/T_{new2}	T_{new2}/T_L	T_{new2}/T_{YQ}	T_{new2}/T_{FM}	T_{new2}/T_{YBH}
1	21	10	58	5-MC	0.797	0.064	0.064	0.287	0.954
				7-MC	0.751	0.028	0.067	0.266	0.789
				9-MC	0.771	0.021	0.088	0.252	0.607
				11-MC	0.755	0.014	0.482	0.260	0.161
2	21	12	111	5-MC	0.571	0.036	0.036	0.138	0.945
				7-MC	0.612	0.019	0.020	0.120	0.958
				9-MC	0.711	0.015	0.020	0.148	0.894
				11-MC	0.858	0.016	0.059	0.158	0.481
3	23	13	140	5-MC	0.723	0.045	0.045	0.304	0.945
				7-MC	0.792	0.026	0.035	0.225	0.927
				9-MC	0.932	0.028	0.057	0.153	0.574
				11-MC	0.944	0.018	0.336	0.129	0.060

Note: T_{new1}, and T_{new2} are the running times of the suggested algorithm using Theorem 8, and Theorem 9, respectively, to detect and remove duplicate *d*-MCs; T_L, T_{YQ}, T_{FM}, and T_{YBH} are the running times of Lin's algorithm (2002), Yan and Qian's algorithm (2007), Forghani-elahabad and Mahdavi-Amiri's algorithm (2014), and the algorithm of Yeh et al. (2015), respectively.

In summary, the developed technique for finding lower capacity bounds is beneficial to solving d-MCs, and the two judging criteria outperforms the traditional comparison method with regard to detecting duplicate d-MCs. What is more, the experimental results firmly support the superiority of the suggested algorithm over the existing algorithms.

5. A case study of LCD monitor delivery

Reliability evaluated during the operation phase of complex technological networks is a key indicator to measure the level of service of the networks. Furthermore, this information can be deemed as a performance criterion to figure out the optimal scheme for network improvement (Kuo and Zuo, 2003; Ramirez-Marquez and Coit, 2005a). As an NP-hard problem, reliability evaluation has long been recognized to be a difficult and challenging task. Developing efficient algorithms for reliability analysis contributes to the quick and accurate demonstration of network performance, and thus is a popular topic to both researchers and practitioners. We have proposed a new d-MC method to evaluate the reliability of a stochastic distribution network, and its performance advantages over the existing methods have already been proved through both theoretical and numerical results. In this section, a practical distribution network, as presented in Fig. 8, is utilized to further illustrate the application of the proposed algorithm.

5.1. Reliability evaluation of an LCD monitor distribution network between China and France

A Chinese manufacturer owning a factory located at Shenzhen city in China produces LCD monitor commodities. LCD monitors are usually utilized in consumer electronics industry. Owing to the excellent quality and price advantage of products, the manufacturer has not only been one of the chief LCD monitor providers for Chinese consumer electronics companies, but also been recognized as the premium supplier by many international consumer electronics enterprises for years. One of its customers is a famous consumer electronics company in France. When confirming an order from the company in France, the manufacturer is responsible for the accurate delivery of LCD monitors. Fig. 8 illustrates the LCD monitor distribution network between China and France, in which the LCD monitor commodity can pass through several transfer centers in different countries.

The manufacturer has gotten an order from the company in France to deliver 1000 pieces of 42 in. LCD monitors. The dimension of each 42 in. LCD monitor is $105.1 \times 73.89 \times 29.1$ (unit: cm^3). During delivery by either ship or truck, the LCD monitors are typically loaded onto TEU (twenty-feet equivalent unit). The size of TEU is $589.8 \times 235.2 \times 238.5$ (unit: cm^3), and can load approximately 146 pieces of 42 in. LCD monitors. Each carrier along routes has multiple available capacities, such as 0, 1, ..., 5 TEU with the probability distribution derived from the carrier's database. The capacity data of carriers along routes are shown in Table 8. For example, $\Pr(x_1 = 2) = 0.029$ implies the probability that the carrier on route e_1 exactly provides 2 TEU per unit of period is 0.029. And, the probability that the carrier on route e_1 can provide more than or equal to 2 TEU per unit of period is 0.974 because $\Pr(x_1 = 2) + \Pr(x_1 = 3) = 0.974$.

Since one TEU can load approximately 146 pieces of 42 in. LCD monitors, the capacity of the distribution network should be more than or equal to 7 TEU in order to load 1000 pieces of 42 in. LCD monitors (i.e. demand level is 7). Therefore, if the manager would like to assess the capability of the network to ensure the delivery of required quantity of LCD monitors, the performance index R_7 can be evaluated in terms of 6-MCs, i.e. $d = 6$. There are 20 MCs in Fig. 8. By using the proposed algo-

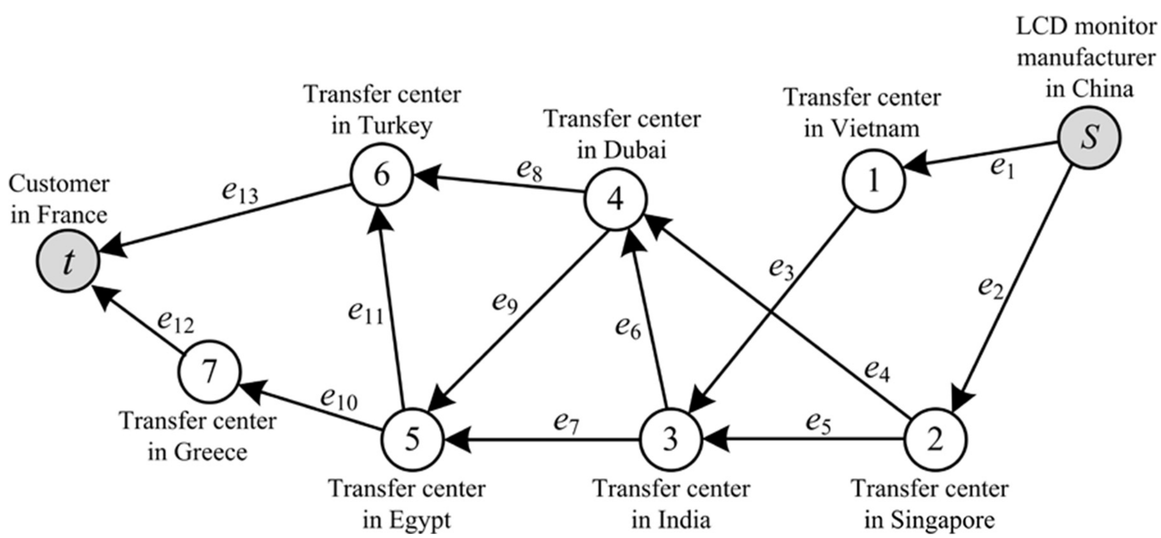

Fig. 8. An LCD monitor distribution network between China and France.

rithm, a total number of 104 6-MCs are found. Based on all 6-MCs and the RSD method shown in Eq. (1), the performance index $R_7 = 0.778660$ is obtained. As a result, the probability that the distribution network in Fig. 8 can successfully deliver 7 TEU of LCD monitor commodities from China to France is 0.778660. Undoubtedly, this value gives the manager valuable information on the capability of the distribution network to accomplish the delivery of required quantity of LCD monitor commodities, and thus can be regarded as a decision criterion. For example, if the probability 0.778660 is below the threshold set by the manager, it means that the performance of the distribution network is unsatisfactory, and thereby it is necessary to improve the network. Otherwise, the performance of the distribution network is desirable.

For comparison, network reliabilities at different demand levels are also computed, and the results are summarized in Table 9. As expected, network reliability decreases as the demand level increases. Furthermore, the data of the last row in Table 9 reveals that the reliability difference between two successive demand levels increases as the demand level increases. When d ranges from 1 to 5, the reliability difference between two successive demand levels (i.e. $R_{d+1} - R_{d+2}$) is not notable, but it becomes prominent from $d = 5$ to $d = 7$. Therefore, $d = 5$ (corresponds to the reliability R_6) is a critical value above which the network reliability sharply decreases.

5.2. Optimal scheme for improving the LCD monitor distribution network

The manager pays close attention to the accurate delivery of LCD monitor commodities. Given that the reliability 0.778660 is below the manager's expectation, and the manager intends to improve the current distribution network by adding new routes (arcs) to it. As a result, the manager needs to determine which routes are the best for network improvement. In addition, because of some objective conditions, only several routes are specified to be the candidates for network improvement. That is, the candidate routes are limited to China to India, Vietnam to Dubai, Dubai to Greece, and Egypt to France, i.e. four candidate routes, and their capacity data are presented in Table 10. Consequently, the optimal network improvement scheme is to determine the best from the four candidate routes, such that their addition to the current network results in the maximal network reliability. To make a comprehensive analysis, three cases are analyzed:

Case 1: One route is added to the existing network.
Case 2: Two routes is added to the existing network.
Case 3: Three routes is added to the existing network.

The reliability of the expanded network is calculated using the proposed algorithm, and the results under different selections for each case are presented in Table 11 by which we state the following observations:

(1) China to India is the best for network improvement if the manager considers to add only one route to the current network. Similarly, China to India and Egypt to France (China to India, Dubai to Greece and Egypt to France) are the optimal routes for network improvement when the manager plans to add two (three) routes to the current network.
(2) If the manager hopes that the reliability of the expanded network marginally exceeds 0.95, the best choice is adding the two routes China to India and Egypt to France to the current network.
(3) It is noteworthy that the route China to India is always involved in the optimal scheme in every case, thus the manager should give top priority to the route China to India whenever considering to improve the existing network.
(4) As expected, the reliability of the expanded network under the optimal scheme increases as the number of added routes increases.

Table 8
Capacity data of routes (arcs) in Fig. 8.

Route	Available capacity (unit: TEU)					
	0	1	2	3	4	5
	Probability					
e_1	0.008	0.018	0.029	0.945	–	–
e_2	0.007	0.009	0.016	0.023	0.032	0.913
e_3	0.005	0.021	0.023	0.951	–	–
e_4	0.006	0.019	0.026	0.949	–	–
e_5	0.011	0.018	0.025	0.946	–	–
e_6	0.015	0.026	0.044	0.915	–	–
e_7	0.007	0.019	0.031	0.042	0.901	–
e_8	0.006	0.014	0.023	0.035	0.922	–
e_9	0.011	0.053	0.936	–	–	–
e_{10}	0.008	0.013	0.026	0.037	0.916	–
e_{11}	0.012	0.056	0.932	–	–	–
e_{12}	0.006	0.019	0.023	0.031	0.921	–
e_{13}	0.007	0.012	0.019	0.023	0.037	0.902

Table 9
Network reliabilities at different demand levels.

d	1	2	3	4	5	6	7
No. of d-MCs	64	140	203	204	140	73	34
R_{d+1}	0.999000	0.996841	0.985919	0.962932	0.902387	0.778660	0.610428
$R_{d+1} - R_{d+2}$	0.002159	0.010922	0.022987	0.060545	0.123727	0.168232	–

Table 10
Capacity data of the candidate routes.

Candidate route	Available capacity (unit: TEU)			
	0	1	2	3
	Probability			
China to India	0.005	0.012	0.026	0.957
Vietnam to Dubai	0.002	0.008	0.013	0.977
Dubai to Greece	0.004	0.009	0.011	0.976
Egypt to France	0.003	0.009	0.015	0.973

Table 11
Network reliabilities under different network improvements.

No. of new routes	The added route(s)	Reliability of the improved network	The difference to the reliability without network improvement	Remark
1	China to India	**0.876436**	**0.097776**	Optimal
	Vietnam to Dubai	0.804974	0.026314	–
	Dubai to Greece	0.801931	0.023271	–
	Egypt to France	0.844690	0.066030	–
2	China to India and Vietnam to Dubai	0.884229	0.105569	–
	China to India and Dubai to Greece	0.902382	0.123722	–
	China to India and Egypt to France	**0.950799**	**0.172139**	Optimal
	Vietnam to Dubai and Dubai to Greece	0.835693	0.057033	–
	Vietnam to Dubai and Egypt to France	0.873018	0.094358	–
	Dubai to Greece and Egypt to France	0.862951	0.084291	–
3	China to India, Vietnam to Dubai and Dubai to Greece	0.917757	0.139097	–
	China to India, Vietnam to Dubai and Egypt to France	0.959055	0.180395	–
	China to India, Dubai to Greece and Egypt to France	**0.971245**	**0.192585**	Optimal
	Vietnam to Dubai, Dubai to Greece and Egypt to France	0.898314	0.119654	–

Note: The reliability and difference marked in bold are the maximal reliability and the biggest difference in each case, respectively.

6. Concluding remarks

With increasing demands for better and more reliable service, the reliability problem has become a major concern in the design of new networks, and the operation and improvement of existing networks. This paper proposes a d-MC based algorithm to evaluate the reliability of a stochastic distribution network that can be regarded as a typical stochastic-flow network where each arc has a random capacity and the corresponding operational reliability. To improve the efficiency of solving d-MCs, a new technique is developed to find lower capacity bounds of arcs which are used to cut down the number of d-MC candidates. Also, a more effective method based on two judging criteria is presented to overcome the drawbacks of the existing methods in detecting duplicate d-MCs. The proposed judging criteria are the first to really find out the underlying reason for the generation of duplicate d-MCs. Both complexity analysis and computational experiments conducted on benchmark networks indicate that the suggested algorithm outperforms the existing methods. Through a practical distribution network related to LCD monitor products, this study not only demonstrates the utility of the proposed algorithm but also discusses the management implications of network reliability. A manager can take network reliability as a decision criterion to determine the optimal network improvement scheme.

For future research, there is still a great potential for extending the suggested algorithm to more practical applications. For instance, the cost associated with distribution activity is also a major concern for logistics providers, and thus the problem of network reliability subject to budget constraint is a practical one worthy of study. Hence, it is meaningful to extend the proposed algorithm to be applicable to the cost and reliability integrated performance evaluation of a distribution network. Furthermore, the distribution network discussed in this paper is in fact a single-commodity stochastic-flow network in which only one type of commodity is transported from the source to the destination. In practice, however, a distribution network usually allows multiple types of commodity to be delivered from the source to the destination simultaneously. Because different types of commodity may consume the arc capacity differently, it is inappropriate to treat the network capacity as the maximal sum of the commodity. Therefore, it is worthwhile to study how to modify the suggested algorithm to evaluate the reliability of multi-commodity stochastic-flow networks, which would enhance applicability of the algorithm.

Acknowledgements

The authors gratefully acknowledge the editor-in-chief, Prof. Jiuh-Biing Sheu, and the anonymous reviewers for their valuable comments and suggestions that have significantly improved the paper. This work is jointly supported by the National Basic Research Program of China (Project No. 2012CB725401), the National Natural Science Foundation of China (Project Nos. 71621001, 71601072, 71601073), the Research Committee of The Hong Kong Polytechnic University (Project No. 87PV), and the Research Institute for Sustainable Urban Development of The Hong Kong Polytechnic University (Project Nos. 1-ZVBX and 1-ZVBY).

Appendix A

A.1. The existing results with respect to the d-MC problem

Lemma 2. *For a d-MC candidate X, if* $M(X) = d$ *and there is a path between the source node and the destination node in* $R^d(V, E, X+0(e_i))$ *for all* $e_i \in U(X)$, *then X is a d-MC.*

Lemma 3. *For a d-MC candidate X with* $M(X) = d$, *if* $|U(X)| = 1$, *then X is a d-MC.*

Lemma 4. *For an MC C, if* $Cap(C) = D$, *every d-MC candidate derived from C is a d-MC.*

Lemma 2 is used to verify whether a d-MC candidate is a d-MC, and Lemmas 3 and 4 describe two special cases in which the verifications are avoidable. Lemma 3 reveals that there is only one unsaturated component in the d-MC. In Lemma 4, Cap$(C) = D$ means the capacity of C is minimal among all MCs.

Lemma 5. *For a d-MC candidate X, if there exists no d-MC candidate* X^*, *such that* $X \leq X^*$, *then X is a d-MC without duplicates.*

Lemma 6. *Let* C_i *and* C_j *be two distinct MCs and X be a d-MC generated from* C_i, *if* $U(X) \subseteq C_j$, *X is also a d-MC generated from* C_j, *i.e., X is a duplicate d-MC.*

Lammas 5 and 6 are utilized to detect duplicate d-MCs. The well-known comparison method is based on Lemma 5, and Yeh's method (2008) relies on Lemma 6. Lemma 5 indicates that each d-MC candidate must be verified by comparing it with all of the other d-MC candidates. It is a time-consuming task to implement the comparison process due to the exponentially growing number of d-MC candidates (Yeh, 2008). Lemma 6, as opposed to Lemma 5**,** provides helpful insight into the fundamental reason for the generation of duplicate d-MCs. Unfortunately, Lemma 6 may work incorrectly under certain condition. Specifically, the usage of Lemma 6 may lead to the loss of non-duplicate d-MCs. Fig. 1 is used as an example to illustrate this point.

Fig. 1 shows that there are 4 MCs in the network: $C_1 = \{e_1, e_2, e_3\}$, $C_2 = \{e_1, e_3, e_4, e_6\}$, $C_3 = \{e_2, e_3, e_4, e_5\}$, and $C_4 = \{e_3, e_5, e_6\}$. It is easy to check that $X = (4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$ is an 8-MC derived from $C_4 = \{e_3, e_5, e_6\}$, and $U(X) = \{e_3, e_5\} \subseteq C_3$. Consequently, $(4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$ is identified by Lemma 6 as a duplicate 8-MC also generated from C_3, and it will be removed from the list of 8-MCs. Actually, however, $X = (4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$ is not an 8-MC (candidate) generated from C_3 because $\sum_{e \in C_3} X(e) = X(e_2) + X(e_3) + X(e_4) + X(e_5) = 9 \neq 8$. Hence, the non-duplicate 8-MC $(4, 3, \underline{\mathbf{3}}, 1, \underline{\mathbf{2}}, \mathbf{3})$ is lost due to the malfunction of Lemma 6.

A.2. Several newly obtained results

The following conclusion is directly from the definition of $L(e_i)$ $(1 \leq i \leq m)$.

Corollary 1. *For an arc* e_i $(1 \leq i \leq m)$, *if* $L(e_i)$ *exists, then* $M(X) = d$, *where* $X = (W_1, W_2, \ldots, W_{i-1}, L(e_i), W_{i+1}, \ldots, W_m)$.

The basic requirement for a capacity vector X to be a d-MC is the satisfaction of the flow demand d, thus the following conclusion is at hand.

Corollary 2. *For a d-MC candidate* $X = (X(e_1), X(e_2), \ldots, X(e_m))$ *derived from MC C, if X is a d-MC, then* $X(e_i) \geq L(e_i)$ *when* $L(e_i)$ *exists for all* $e_i \in C$.

Proof. By the definition of lower capacity bound, we have $M(X) < d$ if $X(e_i) < L(e_i)$ for $e_i \in C$, which is contrary to the definition of d-MC. Hence, the conclusion holds true.□

Grounded on Lemma 3 and Corollary 1, it is trivial to obtain the following conclusion.

Corollary 3. *For an arc e_i $(1 \le i \le m)$, if $L(e_i)$ exists, then $X = (W_1, W_2, \ldots, W_{i-1}, L(e_i), W_{i+1}, \ldots, W_m)$ is a d-MC.*

Corollary 4. *If $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC generated from MC C, then*

$$\sum_{e \in E} X(e) = d + \sum_{e \in E} W(e) - \sum_{e \in C} W(e).$$

Proof. By Lemma 1,

$$\begin{aligned}\sum_{e \in E} X(e) &= \sum_{e \in C} X(e) + \sum_{e \notin C} W(e) \\ &= d + \sum_{e \notin C} W(e) \\ &= d + \sum_{e \in E} W(e) - \sum_{e \in C} W(e). \ \square\end{aligned}$$

Corollary 5. *If $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC generated from MC C, then $U(X) \subseteq C$.*

Proof. It is directly from Eq. (4) in Lemma 1. □

Corollaries 4 and 5 point out the basic properties which a d-MC should satisfy.

A.3. Nomenclature

Demand level d	$0 \le d < D$, a non-negative integer-valued flow demand from the source to the destination
Cut	a cut is a subset of E such that there exists no path from the source node to the destination node after elimination of all its elements from $G(V, E, W)$
Minimal cut	a cut such that none of its proper subsets is a cut
d-MC	a network capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$ is a d-MC if and only if $M(X) = d$, and $M(X+0(e_i)) > d$ for each $e_i \in U(X)$
$Y \ge X$	$(Y_1, Y_2, \ldots, Y_m) \ge (X_1, X_2, \ldots, X_m)$ with $Y_i \ge X_i$ for $i = 1, 2, \ldots, m$
$Y > X$	$(Y_1, Y_2, \ldots, Y_m) > (X_1, X_2, \ldots, X_m)$ with $Y_i \ge X_i$ for $i = 1, 2, \ldots, m$, and $Y_i > X_i$ for at least one i

A.4. Notations

$G(V, E, W)$	a stochastic-flow network with the set of nodes $V = \{1, 2, \ldots, n\}$, where 1 is the source node and n is the destination node, the set of arcs $E = \{e_1, e_2, \ldots, e_m\}$, and $W = (W_1, W_2, \ldots, W_i)$, where $W_i = W(e_i)$ denotes the max-capacity of e_i for $1 \le i \le m$
$G(V, E, X)$	the network corresponds to $G(V, E, W)$ except that W is replaced by X
$R^d(V, E, X)$	the corresponding residual network to $G(V, E, X)$ after sending d units of flow from source node 1 to destination node n
e	an arc
e_i	the i^{th} arc in E
m, n	the number of arcs, and the number of nodes
$X(e_i)$	the current capacity of arc e_i representing the amount of flow allowed to be sent through e_i
$L(e_i)$	lower capacity bound of arc e_i in d-MCs
E^*	$E^* = \{e_i \mid L(e_i) \text{ exists}\}$ which is a subset of E
X	a network capacity vector $X = (X(e_1), X(e_2), \ldots, X(e_m))$
C	a minimal cut
C_i	the ith minimal cut

(continued on next page)

Cap(C)	the capacity of minimal cut C, i.e., Cap(C) = $\sum_{e\in C} W(e)$
$M(X)$	the max-flow of the network under X
$U(X)$	$U(X) = \{e_i \mid X(e_i) < W(e_i)\}$
$W(0_i)$	$W(0_i)$= ($W_1, W_2, \ldots, W_{i-1}, 0, W_{i+1}, \ldots, W_m$), capacity is 0 for e_i and the largest for others
$0(e_i)$	$0(e_i)$ = (0, 0, . . . , 0, 1, 0, . . . , 0), i.e. capacity is 1 for e_i and zero for others
d	demand level for the network
D	$M(W)$, the max-flow of the network under W
p, σ	the number of MCs in the network, and the number of d-MC candidates obtained from an MC, respectively
R_{d+1}	reliability at demand level d+1
$\lvert\bullet\rvert$	the number of elements of $\bullet$, e.g. $\lvert V\rvert$ is the number of nodes in V

References

Ahuja, R.K., Kodialam, M., Mishra, A.K., Orlin, J.B., 1997. Computational investigations of maximum flow algorithms. Eur. J. Oper. Res. 97 (3), 509–542.
Aggarwa, K.K., Misra, K.B., Gupta, J.S., 1975. A simple method for reliability evaluation of a communication system. IEEE Trans. Commun. 23 (5), 563–565.
Alexopoulos, C., 1995. Note on state-space decomposition methods for analyzing stochastic flow networks. IEEE Trans. Commun. 44, 354–357.
Bai, G.H., Zuo, M.J., Tian, Z.G., 2015. Ordering heuristics for reliability evaluation of multistate networks. IEEE Trans. Reliab. 64 (3), 1015–1023.
Ball, M.O., 1993. Computational complexity of network reliability analysis an overview. IEEE Trans. Reliab. 35 (3), 230–239.
Colbourn, C.J., 1987. The Combinatorics of Network Reliability. Oxford University Press, New York, USA.
Chopra, S., 2003. Designing the distribution network in a supply chain. Transp. Res. Part E 39, 123–140.
Cedillo-Campos, M.G., Riva-Canizales, J.L., Bueno-Solano, A., Gonzalez-Feliu, j., García-Alcaraz, J.L., 2014. Reliability in urban freight distribution: a Markovian approach. DYNA 81 (187), 232–239.
Chen, A., Kasikitwiwat, P., Yang, C., 2013. Alternate capacity reliability measures for transportation networks. J. Adv. Transp. 47 (1), 79–104.
Chen, A., Yang, H., Lo, H.K., Tang, W., 2002. Capacity reliability of a road network: an assessment methodology and numerical results. Transp. Res. Part B 36 (3), 225–252.
Doulliez, P., Jamoulle, E., 1972. Transportation networks with random arc capacities. RAIRO 6, 45–60.
Edrissi, A., Nourinejad, M., Roorda, M.J., 2015. Transportation network reliability in emergency response. Transp. Res. Part E 80, 56–73.
Forghani-elahabad, M., Mahdavi-Amiri, N., 2013. On search for all d-MCs in a network flow. Iran J. Oper. Res. 4 (2), 108–126.
Forghani-elahabad, M., Mahdavi-Amiri, N., 2014. A new efficient approach to search for all multi-state minimal cuts. IEEE Trans. Reliab. 63 (1), 154–166.
Ho, W., Emrouznejad, A., 2009. Multi-criteria logistics distribution network design using SAS/OR. Expert Syst. Appl. 36 (3), 7288–7298.
Jane, C.C., Lin, J.S., Yuan, J., 1993. On reliability evaluation of a limited-flow network in terms of minimal cutsets. IEEE Trans. Reliab. 42 (3), 354–361.
Jane, C.C., Laih, Y.W., 2008. A practical algorithm for computing multi-state two-terminal reliability. IEEE Trans. Reliab. 57 (2), 295–302.
Jane, C.C., Laih, Y.W., 2010. Computing multi-state two-terminal reliability through critical arc states that interrupt demand. IEEE Trans. Reliab. 59 (2), 338–344.
Jane, C.C., 2011. Performance evaluation of logistics systems under cost and reliability considerations. Transp. Res. Part E 47 (2), 130–137.
Kuo, W., Zuo, M.J., 2003. Optimal Reliability Modeling: Principles and Applications. Wiley, New York, NY.
Lin, J.S., 1998. Reliability evaluation of capacitated-flow networks with budget constraints. IIE Trans. 30 (12), 1175–1180.
Lin, Y.K., 2002. Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs. Reliab. Eng. Syst. Saf. 75 (1), 41–46.
Lin, Y.K., 2007. Performance evaluation for the logistics system in case that capacity weight varies from arcs and types of commodity. Int. J. Prod. Econ. 107 (2), 572–580.
Lin, Y.K., 2009. System reliability evaluation for a multistate supply chain network with failure nodes using minimal paths. IEEE Trans. Reliab. 58 (1), 34–40.
Lin, Y.K., Yeh, C.T., Huang, C.F., 2013. Reliability evaluation of a stochastic-flow distribution network with delivery spoilage. Comput. Ind. Eng. 66 (2), 352–359.
Niu, Y.F., Xu, X.Z., 2012. Reliability evaluation of multi-state systems under cost consideration. Appl. Math. Model. 36 (9), 4261–4270.
Niu, Y.F., Lam, Willianm H.K., Gao, Z.Y., 2014. An efficient algorithm for evaluating logistics network reliability subject to distribution cost. Transp. Res. Part E 67, 175–189.
Nagurney, A., Yu, M., Floden, J., Nagurney, L.S., 2014. Supply chain network competition in time-sensitive markets. Transp. Res. Part E 70, 112–127.
Nagurney, A., Saberi, S., Shukla, S., Floden, J., 2015. Supply chain network competition in price and quality with multiple manufacturers and freight service providers. Transp. Res. Part E 77, 248–267.
Peng, P., Snyder, L.V., Lim, A., Liu, Z.L., 2011. Reliable logistics networks design with facility disruptions. Transp. Res. Part B 45, 1190–1211.
Ramirez-Marquez, J.E., Coit, D.W., 2005a. Composite importance measures for multistate systems with multistate components. IEEE Trans. Reliab. 54 (3), 517–529.
Ramirez-Marquez, J.E., Coit, D.W., 2005b. A Monte-Carlo simulation approach for approximating multi-state two-terminal reliability. Reliab. Eng. Syst. Saf. 87 (2), 253–264.
Satitsatian, S., Kapur, K.C., 2006. An algorithm for lower reliability bounds of multi-state two-terminal networks. IEEE Trans. Reliab. 55 (2), 199–206.
Soh, S., Rai, S., 1993. Experimental results on preprocessing of path/cut term in the sum of disjoint products technique. IEEE Trans. Reliab. 42 (1), 24–33.
Sheu, J.B., 2006. A novel dynamic resource allocation model for demand-responsive city logistics distribution operations. Transp. Res. Part E 42 (6), 445–472.
Sheu, J.B., 2007. An emergency logistics distribution approach for quick response to urgent relief demand in disasters. Transp. Res. Part E 43 (6), 687–709.
Sheu, J.B., 2010. Dynamic relief-demand management for emergency logistics operations under large-scale disasters. Transp. Res. Part E 46 (1), 1–17.
Soltani-Sobh, A., Heaslip, K., El Khoury, J., 2015. Estimation of road network reliability on resiliency: an uncertain based model. Int. J. Disaster Risk Reduction 14, 536–544.
Soltani-Sobh, A., Heaslip, K., Scarlatos, P., Kaisar, E., 2016a. Reliability based pre-positioning of recovery centers for resilient transportation infrastructure. Int. J. Disaster Risk Reduction 19, 324–333.
Soltani-Sobh, A., Heaslip, K., Stevanovic, A., El Khoury, J., Song, Z., 2016b. Evaluation of transportation network reliability during unexpected events with multiple uncertainties. Int. J. Disaster Risk Reduction 17, 128–136.
Shier, D., 1991. Network Reliability and Algebraic Structures. Clarendon Press, New York, USA.
Tsao, Y.C., Lu, J.C., 2012. A supply chain network design considering transportation cost discounts. Transp. Res. Part E 48, 401–414.
Whicker, L., Bernon, M., Templar, S., Mena, C., 2009. Understanding the relationships between time and cost to improve supply chain performance. Int. J. Prod. Econ. 121 (2), 641–650.
Yeh, C.T., Lin, Y.K., Huang, C.F., 2014. A reliability indicator to measure a stochastic supply chain network with transportation damage and limited production capacity. IIE Trans. 46 (10), 1066–1078.
Yeh, W.C., 2002. A new approach to the d-MC problem. Reliab. Eng. Syst. Saf. 77 (2), 15–19.

Yeh, W.C., 2004. A simple MC-based algorithm for evaluating reliability of stochastic-flow network with unreliable nodes. Reliab. Eng. Syst. Saf. 83 (1), 47–55.
Yeh, W.C., 2008. A fast algorithm for searching all multi-state minimal cuts. IEEE Trans. Reliab. 57 (4), 581–588.
Yeh, W.C., Bae, C., Huang, C.L., 2015. A new cut-based algorithm for the multi-state flow network reliability problem. Reliab. Eng. Syst. Saf. 136 (1), 1–7.
Yeh, W.C., 2015. An improved sum-of-disjoint-products technique for symbolic multi-state flow network reliability. IEEE Trans. Reliab. 64 (4), 1185–1193.
Yan, Z., Qian, M., 2007. Improving efficiency of solving *d*-MC problem in stochastic-flow network. Reliab. Eng. Syst. Saf. 92 (1), 30–39.
Yu, M., Nagurney, A., 2013. Competitive food supply chain networks with application to fresh produce. Eur. J. Oper. Res. 224 (2), 273–282.
Yang, X., 2013. A review of distribution related problems in logistics and supply chain research. Int. J. Supply Chain Manage. 2 (4), 1–8.
Zuo, M.J., Tian, Z., Huang, H.Z., 2007. An efficient method for reliability evaluation of multistate networks given all minimal path vectors. IIE Trans. 39, 811–817.
Zio, E., 2009. Reliability engineering: old problems and new challenges. Reliab. Eng. Syst. Saf. 94, 125–141.