Applied Mathematical Modelling 000 (2015) 1–9

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

An improved algorithm for finding all upper boundary points in a stochastic-flow network

Majid Forghani-elahabad [a], Nezam Mahdavi-Amiri [b,*]

[a] Universidade Federal do ABC - UFABC, Santo André, SP, Brazil
[b] Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran

A R T I C L E I N F O

Article history:
Received 15 June 2014
Revised 30 September 2015
Accepted 5 October 2015
Available online xxx

Keywords:
Reliability
Stochastic-flow network
d-MinCut (*d-MC*)
Sum of disjoint products method

A B S T R A C T

The *d*-MinCut (*d-MC*) problem has been extensively studied in the past decades and various algorithms have been proposed. The existing algorithms often consist of two general stages, finding all the *d-MC* candidates and testing each candidate for being a *d-MC*. To find all the *d-MC* candidates, a system of equations and inequalities should be solved. Here, we propose a novel efficient algorithm to solve the system. Then, using our new results, an improved algorithm is proposed for the *d-MC* problem. Computing its complexity, we show the proposed algorithm to be more efficient than the other existing algorithms with respect to the number of minimal cuts. A medium-size network example is worked through to show how effectively the algorithm finds all the *d-MC*s among the set of all possible candidates in the network. Moreover, it is shown how to compute the system reliability of the example using the sum of disjoint products method. Finally, in order to illustrate the efficacy of using the new presented techniques, computational comparative results on an extensive collection of random test problems are provided in the sense of performance profile introduced by Dolan and More.

1. Introduction

After World War I, the reliability was measured as the number of accidents per hour of flight time for one-, two-, and four-engine airplanes [1]. After that, network reliability theory has extensively been applied to a variety of real-world systems such as power transmission and distribution system [2], computer and communication [3,4], transportation [5], etc. The network reliability problem is NP-hard [6].

System reliability, the probability that the maximum flow of the network is more than a demand level d, can be computed in terms of either upper boundary points for demand level d, called d-MinCuts (*d-MC*s) [7–13], or lower boundary points for demand level d, called d-MinPaths (*d-MP*s) [14–16]. Since the number of minimal cuts (*MC*s) is usually less than the number of minimal paths (*MP*s) [17], working on *MC*s is preferred. The search for all *d-MC*s is an NP-hard problem [18].

Introducing the notion of *d-MC* candidate, Jane et al. [9] proposed an algorithm that first finds all the *d-MC* candidates obtained from each *MC* and then checks every candidate for being a *d-MC*. Since then, most algorithms [7,8,10–13] are composed of two general stages: finding all the *d-MC* candidates by an implicit enumeration method and verifying every candidate via a checking procedure for being a *d-MC*.

* Corresponding author. Tel.: +982166005117; fax: +982166165607.
E-mail addresses: m.forghani@ufabc.edu.br, forghanimajid@gmail.com (M. Forghani-elahabad), nezamm@sharif.edu, nezamm@sina.sharif.edu (N. Mahdavi-Amiri).

http://dx.doi.org/10.1016/j.apm.2015.10.004
S0307-904X(15)00637-X/© 2015 Elsevier Inc. All rights reserved.

Lin [10] showed that the maximal elements of all the obtained *d-MC* candidates from all the *MC*s were equal to the set of all the *d-MC*s and then proposed an algorithm that uses a comparative method to specify the set of maximal elements, *d-MC*s, among all the *d-MC* candidates. Proving some new results, Yeh [12] presented an efficient algorithm that employs the max-flow algorithm and the residual network [19] to check every *d-MC* candidate for being a *d-MC*. Yan and Qian [11] proved new results to decrease the number of the obtained *d-MC* candidates, find some *d-MC*s without the need for testing and eliminate some duplicate *d-MC*s. Then, they proposed an improved algorithm, which is more efficient than the proposed algorithms in [9,10,12] (see [11] for a comparative study). Later, Yeh [13] presented some new results, and for the first time proposed an algorithm to avoid the production of duplicate *d-MC*s. Considering the budget constraints, Forghani and Mahdavi-Amiri [20] proposed an effective algorithm to find all the *d-MC*s satisfying the budget constraints. Presenting new results as well as a new data structure, Forghani and Mahdavi-Amiri [8] proposed an efficient algorithm and showed the algorithm to be more efficient than the ones given in [9–13]. It should be noted that some of the presented results in the literature are not working correctly (see [7]).

However, the complexity of the algorithms in [7–13] have not been investigated with respect to solving the required system for finding *d-MC* candidates. Here, we first propose a novel efficient algorithm, Algorithm 1, to solve a Diophantine system appearing in the *d-MC* problem. Then, the correctness and complexity results of Algorithm 1 (Lemma 1 and Theorem 1) are given. Afterwards, a new result (Lemma 2) is presented in order to decrease the number of operations in finding all the *d-MC*s. Thereafter, making use of our newly established results, an improved algorithm, Algorithm 2, is proposed for the *d-MC* problem. Computing the time complexity of our proposed algorithm, it is shown to be more efficient than the ones in [8–13]. Moreover, using the performance profile on the numerical results obtained over randomly generated test problems of Dolan and Moré [21], we show the practical efficiency of our proposed algorithm in comparison with other existing algorithms.

The remainder of our work is organized as follows. Section 2 describes the required notations, nomenclature and assumptions. Some preliminary explanations, a novel approach to find all the *d-MC* candidates, and some results on the *d-MC* problem are stated in Section 3. In Section 4, an improved algorithm is proposed. An analysis of the computational complexity of the algorithm is provided to demonstrate the efficiency of our algorithm. Moreover, a medium-size network is employed to exemplify the capability of the algorithm for finding all the *d-MC*s in such networks, and the system reliability of the example is computed using the sum of disjoint products method. In Section 5, efficiency comparisons are made using the performance profile of Dolan and Moré [21] on the results obtained over randomly generated test problems. Finally, we conclude in Section 6.

2. Notations, nomenclature and assumptions

2.1. Notations

$G(N, A, M)$	a stochastic-flow network with the set of nodes $N = \{1, 2, \ldots, n\}$, the set of arcs $A = \{a_i \mid 1 \le i \le m\}$, and $M = (M_1, M_2, \ldots, M_m)$ with $M_i = M(a_i)$ denoting the max-capacity of a_i, for $1 \le i \le m$.
s, t	the source and sink nodes in $G(N,A,M)$, $s, t \notin N$.
n, m	the number of nodes in N and arcs in A, respectively.
$X(a_i)$	the capacity level of arc a_i under the state vector $X = (x_1, x_2, \ldots, x_m)$.
$e_i = 0(a_i)$	a system-state vector in which the capacity level is 1 for a_i and 0 for other arcs.
C_i	the ith MC in $G(N,A,M)$.
$G(N, A, X)$	the corresponding network to $G(N,A,M)$ with current state vector $X = (x_1, x_2, \ldots, x_m)$.
$R(N, A, X^d)$	the corresponding residual network to $G(N, A, X)$ after sending d units of flow from node s to node t.
$V(X)$	the max-flow from node s to node t in $G(N,A,X)$.
ν	$V(M)$, the max-flow from node s to node t in $G(N,A,M)$.
$U(X)$	$U(X) = \{a \in A \mid X(a) < M(a)\}$ is the set of unsaturated arcs in $G(N,A,X)$.
$K_{C_i}(X)$	the capacity of MC, C_i, under vector $X = (x_1, x_2, \ldots, x_m)$; i.e., $K_{C_i}(X) = \Sigma_{a_k \in C_i} x_k$.
$\lvert\cdot\rvert$	the number of elements; e.g., $\lvert N\rvert$ is the number of nodes in N.

2.2. Nomenclature

Cut: A cut is a subset of A, in which there is no path from the source node s to the sink node t after elimination of all its arcs from $G(N, A, M)$.

Minimal Cut (*MC* **)**: A cut so that none of its proper subsets is a cut.

Demand level *d*: $0 \le d < \nu$ is a non-negative integer-valued flow or stress requirement for a given network flow.

System reliability: If d is a constant, then the system reliability, R_d, is $Pr\{X \mid V(X) > d\}$. If d is a random variable with distribution π_d, then the system reliability, R_d, is equal to $Pr\{X \mid V(X) > d\}.\pi_d$.

2.3. Assumptions

(1) The capacity of each arc $a_i \in A$ is a non-negative integer-valued random number less than or equal to M_i.
(2) The capacities of different arcs are statistically independent.
(3) The flow in $G(N, A, M)$ satisfies the flow conservation law [19].