Assessing the performance of smart grid communication networks under both time and budget constraints

Majid Forghani-elahabad,
Center of Mathematics, Computing, and Cognition - Federal University of ABC, Santo André, SP, Brazil

The smart grid concept has emerged to address the existing problems in the traditional electric grid, which has been functioning for more than a hundred years. The most crucial difference between traditional grids and smart grids is the communication infrastructure applied to the latter. However, coupling between these networks can increase the risk of significant failures. Hence, assessing the performance of the smart grid communication networks is of great importance and thus is considered here. As transmission time and cost play essential roles in many real-world communication networks, both time and budget constraints are considered in this work. To evaluate the performance of communication networks, we assume that the data is transmitted from a source to a destination through a single path. We propose an algorithm that computes the exact probability of transmitting $d$ units of data from the source to the destination within $T$ units of time and the budget of $b$. The algorithm is illustrated through a benchmark network example. The complexity results are also provided. A rather large-size benchmark, that is, Pan European Center of Mathematics, Computing, and Cognition - Federal University of ABC, Santo André, SP, Brazil

A communication network in a smart grid is a network of networks that may utilize several different communication technologies supported by two main communications media, that is, wired and wireless, to be expanded deep into the distribution grids [1], [2]. Thus, the communication network will be responsible for gathering and routing data, monitoring every node, and acting upon the received data. On the one hand, wireless communications have some advantages over wired technologies, such as low-cost infrastructure and ease of connection to difficult or inaccessible areas. However, the nature of the transmission path may cause the signal to attenuate [3]. On the other hand, wired solutions generally do not have interference problems, and their functions are not dependent on batteries, as wireless solutions often do [2].

A communication network is usually modeled in one of the two following ways: (1) The network components have a binary behavior, i.e., they can only have a fully working (operational) or wholly failed state. In this case, the network also has two states, and they are called binary-state networks. (2) The components, and consequently the network, can have more than two states. In this case, the network is called a stochastic-flow network, or a multistate flow network (MFN) [4]. Generally, to model a smart grid communication network as an MFN, all the components in the network that should communicate with the other ones such as smart meters, appliances, or control centers can be considered as a node, and each communication link between two nodes can be considered as an arc.

In a communication network, we may have one or more of the three kinds of two-way communication: (1) two-terminal, (2) $k$-terminal, and (3) all-terminal communication. The two-terminal communication is a case in which two specific components communicate with each other, like the communication between each smart meter and the corresponding center. The $k$-terminal communication is a case in which a...
component communicates with $k$ other components, like the communications between appliances at home and a smart meter. The all-terminal communication is a case where all the network components communicate to each other, like the communications between substations in a smart grid. The focus of this work is on the first case, that is, the two-terminal communication.

Several indexes have been proposed in the literature to assess the performance of the MFNs such as communication networks, transportation networks, and so forth. System reliability is one of the most attractive indexes and indeed of great importance. Many researchers have worked on the issues related to the reliability and Quality of Service (QoS) of smart grid communication networks. Wang et al. [5] proposed a reliability evaluation approach to analyze the reliability of Wide-Area Measurement Systems (WAMS). The method incorporates Markov modeling and state enumeration techniques and covers the backbone communication network in WAMS. Gungor et al. [3] discussed communication technologies and requirements for smart grids and presented some QoS mechanisms for smart grids. The authors also introduced some standards and provided a better understanding of the smart grid and related technologies.

Generally, the reliability is considered as the ability to execute a defined function under specified conditions for a known period of time [6]. Availability in a repairable communication network or reliability in a non-repairable network can be measured as the fraction of time that a specific service is available, as the fraction of data packets that successfully attain the planned destination, or generally as the probability of the existence of a demanded connection [7]. An important issue in the smart grid communication network, from the point of view of quality management, is lessening the transmission time or sending data within a given time. Therefore, many researchers have considered the time constraints in assessment of such networks [8], [9], [10], [11]. Furthermore, transmission cost and limited budget are other key issues that play important roles in the real-world systems. Therefore, the two-terminal reliability index in this work, denoted by $R_{(d,T,b)}$, is defined as the probability of transmitting $d$ units of data from a source to a destination through a single path in the network within $T$ units of time and budget $b$. Many algorithms have been proposed in the literature to compute the network reliability with or without the time and budget constraints which are usually based on the minimal cuts [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24] or minimal paths [4], [8], [9], [10], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]. Although many studies have been made so far in this regard, the problem of assessing the reliability of communication networks is a well-known NP-hard combinatorial problem [37], and hence the research continues. This work proposes an algorithm to compute the exact amount of $R_{(d,T,b)}$ for a smart grid communication network. We illustrate the algorithm through an example and compute the complexity results. Then, several numerical results are conducted through known large-size benchmark and one thousand randomly generated test problems to show the practical superiority of our proposed algorithm in comparison with some available algorithm in the literature. The rest of this work is organized as follows. Section II states the required notations and basic results on the problem. The proposed algorithm is given in Section III. We provide the illustrative example and complexity results in Section IV. Section V conducts several numerical results to show the practical efficiency of the proposed algorithm. Finally, we conclude the work in Section VI.

## II. NETWORK MODEL AND BASIC RESULTS

### A. Preliminaries

An architecture of a smart grid communication network is depicted in Fig. 1 taken from [38]. One observes that the corresponding communication network to the smart grid given in Fig. 1 is the one shown in Fig. 2. This network has 10 nodes and 17 arcs which can be considered a multistate flow network (MFN). It shows how one can study a smart grid communication network (SGCN) as an MFN. From now on, we say network instead of SGCN or MFN.

Let $G(N, A, M, L, C)$ be a network with the set of nodes $N = \{1, 2, \ldots, n\}$, the set of arcs $A = \{a_1, a_2, \ldots, a_m\}$, the maximum capacity vector $M = (M_1, M_2, \ldots, M_m)$ in which $M_i$ is an integer-valued number and denotes the maximum capacity of arc $a_i$, for $i = 1, 2, \ldots, m$, the lead time vector $L = (l_1, l_2, \ldots, l_m)$ in which $l_i$ is an integer-valued number and denotes the lead time of arc $a_i$, for $i = 1, 2, \ldots, m$, and the cost vector $C = (c_1, c_2, \ldots, c_m)$ with $c_i$ denoting the transmission cost of $a_i$ for sending each unit of data, for $i = 1, \ldots, m$. Hence, $n$ and $m$ are respectively the number of nodes and arcs in the network. Let also the nodes 1 and $n$ be respectively the source and destination (sink) nodes. For example, Fig. 3 depicts a network with $N = \{1, 2, 3, 4, 5\}$ and $A = \{a_1, a_2, \ldots, a_8\}$. One may consider $M = (5, 3, 4, 3, 2, 4, 5, 3)$, $L = (2, 2, 3, 1, 2, 2, 3, 1)$ and $C = (1, 2, 1, 3, 2, 3, 4)$ as the maximum capacity, lead time and transmission cost vectors in this network. Hence, for instance, at any time no more than 5 units of data can be sent simultaneously through arc $a_1$ as $M_1 = 5$. Moreover, as $l_2 = 2$, it takes two units of time to transmit data from the initial node of $a_1$, that is, 1, to its terminal node, that is, 2. Also, transmitting any units of data on $a_1$ will cost 1 unit of currency as $c_1 = 1$. Although the lead time and transmission cost of any arc is fixed, the capacities of the arcs are not fixed and may vary due to the operations activities, failures, maintenance needs, and so forth. Hence, the current capacity of arc $a_i$ is denoted with $x_i$ which takes values from $\{0, 1, \ldots, M_i\}$ according to a given probability distribution function, for $i = 1, 2, \ldots, m$, and thus $X = (x_1, x_2, \ldots, x_m)$ denotes the current capacity vector of the network. For instance, $X^* = (3, 3, 4, 2, 4, 4, 3)$ can be a current SSV for Fig. 3 under which for example a maximum of 4 units of data can be sent through arc $a_3$ per unit of time. Moreover, it takes $l_3 = 3$ units of time to transmit these 4 units of data from node 1 to node 3 through this arc. Now, if we want transmit 10 units of data from nodes 1 to 3 on $a_3$ under $X^*$, it needs a budget of $b = 10 \times 2 = 20$ and will take $3 + \lceil 10/4 \rceil = 6$ units of time, where $\lceil \alpha \rceil$ is the smallest integer number not less than...
α. Starting the transmission from node 1, \( x_3 = 4 \) units of data can be pumped into \( a_3 \) per unit of time and would take three units of time for the first data arrives at the terminal node. Afterward, \( x_3 = 4 \) units of data will be pumped out from the terminal node each unit of time, and consequently it will take \( \lceil 10/4 \rceil = 3 \) units of time since then until all of the required data is transmitted to the terminal node. Therefore, it takes totally \( 3 + \lceil 10/4 \rceil = 6 \) units of time to complete the transmission. As a result, the transmission time and cost for sending \( d \) units of data on \( a_i \) with the current capacity of \( x_i \), lead time of \( l_i \) and transmission cost of \( c_i \) are respectively equal to

\[
t = l_i + \lceil \frac{d}{x_i} \rceil \quad \text{and} \quad c = d \times c_i. \tag{1}
\]

Pan-European-Network A path is a set of adjacent arcs through which the data can be transmitted from the source node to the destination node, and a minimal path (MP) is a path whose proper subsets are not path anymore. For instance, in Fig. 3 \( P = \{a_1, a_4, a_2, a_3, a_8\} \) is a path from node 1 to node 5 but not an MP because its proper subset, that is, \( P' = \{a_3, a_8\} \), is still a path. However, \( P' = \{a_3, a_8\} \) is an MP. Without loss of generality, let \( P_j = \{a_1, a_2, \ldots, a_{n_j}\} \) be a path in the network with current SSV of \( X = (x_1, x_2, \ldots, x_m) \). The capacity of \( P_j \) under \( X \) is equal to

\[
KP_j(X) = \min\{x_1, x_2, \ldots, x_{n_j}\},
\]

the lead time of this path is equal to

\[
LP_j = \sum_{r=1}^{n_j} l_r,
\]

and finally the transmission cost of this path per unit of data is equal to

\[
CP_j = \sum_{r=1}^{n_j} c_r.
\]

For instance, considering \( X^* = (3, 3, 4, 1, 2, 1, 2, 2) \), \( L = (2, 2, 3, 1, 2, 2, 3, 1) \) and \( C = (1, 2, 2, 1, 3, 2, 3, 4) \) for the given network in Fig. 3, for \( P_1 = \{a_1, a_4, a_7\} \), one has \( KP_1(X^*) = 1 \), \( LP_1 = 6 \) and \( CP_1 = 5 \).

**Lemma 1.** If \( P \) and \( P' \) are two paths in a network such that \( P \subset P' \), then:
1) The lead time of \( P' \) is greater than \( P \),
2) the transmission cost of \( P' \) is greater than \( P \), and
3) the capacity of \( P' \) under any SSV of \( X \) is less than \( P \).

**Proof.** As \( P' \) contain all the arcs of \( P \) and may have some other arc that does not belong to \( P \), the result directly is deduced from the definitions.

**Proposition 1.** For any path \( P \) in a network there is an MP, say \( P_1 \), such that \( P_1 \subset P \).

**Proof.** If \( P \) is an MP, then there is nothing left to prove. If not, then it has a proper subset \( P_1 \), which is still a path. If \( P_1 \) is an MP, the proof is complete. If not, one may repeat this argument. As the number of arcs in any path is limited, this repetition stops at some MP, completing the proof.

The lemma and proposition above show that the paths with minimum transmission cost, maximum capacity, or best lead time are indeed among the MPs of the network. Therefore, we turn our attention to the MPs in the rest of this work.

**B. Basic results**

Here, some results are given based on which an algorithm is proposed in the following section to address the problem.

Let \( P_1, P_2, \ldots, P_q \) are all the minimal paths in the network, and hence \( q \) is the number of MPs. Following a similar argument, which was used for an arc in the preceding section, one can conclude that the transmission time to send \( d \) units of data through MP, \( P_j = \{a_{j_1}, a_{j_2}, \ldots, a_{j_{r_j}}\} \) under the SSV of \( X \) is equal to

\[
\xi(d, X, P_j) = LP_j + \left\lceil \frac{d}{KP_j(X)} \right\rceil.
\]

Similarly, the transmission cost of sending \( d \) units of data through \( P_j \) is equal to

\[
\beta(d, P_j) = d \times CP_j + d \sum_{r=1}^{r_j} c_{j_r}.
\]

For example, considering \( X^* = (3, 3, 4, 1, 2, 1, 2, 2) \), \( L = (2, 2, 3, 1, 2, 2, 3, 1) \) and \( C = (1, 2, 2, 1, 3, 2, 3, 4) \), the transmission time and cost to send \( d = 10 \) units of data through \( P_1 = \{a_1, a_4, a_7\} \) in Fig. 3 under SSV of \( X^* \) are \( \xi(10, X, P_1) = LP_1 + \left\lceil \frac{10}{KP_1(X^*)} \right\rceil = 6 + \left\lceil \frac{10}{1} \right\rceil = 16 \) and \( \beta(d, P_1) = 10CP_1 = 50 \). It is obvious that if \( \beta(d, P_j) > b \), for \( j = 1, 2, \ldots, q \), one cannot transmit \( d \) units of data through a single MP from the source to the destination. Therefore, in this work, we assume that \( \min\{\beta(d, P_j) \mid j = 1, \ldots, q\} \leq b \).

**Lemma 2.** Assume that \( X \) and \( Y \) are two SSVs such that \( X \leq Y \). Then, \( \xi(d, X, P_j) \geq \xi(d, Y, P_j) \), for any demand level of \( d \) and any \( j = 1, 2, \ldots, q \).

**Proof.** Let \( d \) be any given demand level and \( P_j \) any MP in the network. As \( X \leq Y \), one has \( K P_j(X) \leq KP_j(Y) \), and according to Eq. (2) it is concluded that \( \xi(d, X, P_j) \geq \xi(d, Y, P_j) \).

On the one hand, for sending \( d \) units of data from the source to the destination, the transmission time depends on the current SSV while the transmission cost does not. If for an SSV of \( X \) and an MP, say \( P_j \), we have \( \xi(d, X, P_j) \leq T \) but \( d \times CP_j > b \), then it is vivid that one cannot transmit \( d \) units of data through \( P_j \) within budget of \( b \). On the other hand, the data should be sent through a single MP, and that it is always preferable to have the minimum possible transmission time. Therefore, one can integrate the transmission time and cost and define the following function on SSV \( X \) related to transmitting \( d \) units of data from the source node to the destination node through a single MP.

\[
\Xi(d, X) = \min\{\xi(d, X, P_j) \mid \beta(d, P_j) \leq b, \text{ for } j = 1, 2, \ldots, q\}
\]

(4)

Now, one observes that for any SSV of \( X \) with \( \Xi(d, X) \leq T \), there is at least one MP through which \( d \) units of data can be transmitted within \( T \) units of time and budget of \( b \).

**Theorem 1.** If \( X \) and \( Y \) are two SSVs such that \( X \leq Y \), then, \( \Xi(d, X) \geq \Xi(d, Y) \).

**Proof.** It is a direct result of Lemma 2 and Eq. (4).

We recall that the main aim in this work is to assess the performance of a network, that is, to compute the exact network reliability, which is denoted by \( R_{(d,T,b)} \) and is the probability of transmitting \( d \) units of data from a source node to a destination node through a single MP within the time of \( T \) and budget of \( b \). Letting \( \Omega_d = \{X \leq M \mid \Xi(d, X) \leq T\} \), one observes that \( R_{(d,T,b)} = \Pr(X \mid X \in \Omega_d) \). Now, let \( \Omega_{d,\min} = \{X^1, X^2, \ldots, X^q\} \) be the set of all the minimal vectors in \( \Omega_d \) and \( E_r = \{X \mid X \geq X^r\} \), for \( r = 1, 2, \ldots, q \). From Theorem 1 one sees that \( \Omega_d = \cup_{1 \leq r \leq q} E_r \). Therefore, the system reliability can be computed by using some exact methods such as the inclusion-exclusion principle and the sum of disjoint products. For example, using the inclusion-exclusion method, we have

\[
R_{(d,T,b)} = \Pr(\cup_{r=1}^{q} E_r) = \sum_{r=1}^{q} \Pr(E_r) - \sum_{j=2}^{q-1} \sum_{r=1}^{j-1} \Pr(E_r \cap E_j) + \ldots + (-1)^{q+1} \Pr(\cap_{r=1}^{q} E_r),
\]

(5)

where

\[
\Pr(E_r) = \sum_{X \in E_r} \Pr(X), \text{ and } \Pr(X) = \prod_{i=1}^{m} \Pr(x_i).
\]

Now that we know how to compute the network reliability from the set of \( \Omega_d \), the rest is to determine such a set discussed in the next section.

**III. THE PROPOSED ALGORITHM**

We start the discussion here with two definitions.

**Definition 1.** An SSV of \( X \) is called a \((d, T, b)\)-MP candidate if \( \Xi(d, X) \leq T \) (see Eq. (4)).

**Definition 2.** An SSV of \( X \) is called a (real) \((d, T, b)\)-MP if \( \Xi(d, X) \leq T \) and \( \Xi(d, Y) > T \) for any SSV of \( Y < X \).
Lemma 3. Every (real) \((d, T, b)\)-MP is a candidate, and if \(\Omega_d\) is the set of all the \((d, T, b)\)-MP candidate, then \(\Omega_{d, \min}\) is the set of all the \((d, T, b)\)-MPs.

Proof. The first part is clear. Let \(X \in \Omega_{d, \min}\). Assume that \(Y < X\) is an SSV. If \(\exists (d, Y) \leq T\), then \(Y \in \Omega_d\) which contradicts with \(X\) being in \(\Omega_{d, \min}\). Hence, \(\exists (d, Y) > T\), and so \(X\) is a (real) \((d, T, b)\)-MP. Now, let \(X\) be a \((d, T, b)\)-MP. Thus, there is no \(Y < X\) with \(\exists (d, Y) \leq T\), and so \(X \in \Omega_{d, \min}\). □

This lemma shows that one needs to find the minimal vectors among all the \((d, T, b)\)-MP candidates. An elementary approach is to find all the candidates and then determine the minimal vectors by comparing all the candidates. However, both finding the candidates and comparing them are very expensive and time-consuming operations. Assume that the data is transmitting through one MP. Now, let \(X\) be a \((d, T, b)\)-MP. Thus, there is no \(Y < X\) with \(\exists (d, Y) \leq T\), and so \(X \in \Omega_{d, \min}\).

\[\xi(d, X, P_j) = LP_j + \left(\frac{d}{\alpha_j}\right) \leq LP_j + T - LP_j = T.\]

Hence, as \(CP_j \leq \frac{b}{\alpha_j}\), one can send \(d\) units of data through this MP within time \(T\) and budget \(b\). Now, assume that \(Y\) is another SSV such that \(K P_j(Y) = \alpha > \alpha_j\). In this case, we have

\[\xi(d, Y, P_j) = LP_j + \left(\frac{d}{\alpha}\right) > LP_j + T - LP_j = T.\]

Thus, it is not possible to do the same under \(Y\) which completes the proof. □

As a results of this lemma and according to the definition of the capacity of an MP, one can set a vector corresponding to every MP as follows.

Setting the minimum SSV corresponding to an MP:
Suppose that \(P_j\) is an MP such that \(CP_j \leq \frac{b}{\alpha_j}\) and \(LP_j < T\) and let \(\alpha_j = \left\lfloor \frac{d}{T - LP_j} \right\rfloor\). If \(\alpha_j \leq K P_j(M)\), then set the vector \(X_{P_j} = (x_1, x_2, \cdots, x_m)\) as follows.

\[\alpha_j \text{ if } a_i \in P_j, \]  
\[0 \text{ if } a_i \notin P_j.\]  

The first condition to check is whether this vector is less than or equal to \(M\). In other words, if \(\alpha_j > K P_j(M)\), then it is impossible to have such a vector as an SSV for the network. It is why the condition of \(\alpha_j \leq K P_j(M)\) is necessary for the above approach. According to Lemma 4 one observes that \(\xi(d, X_{P_j}, P_j) \leq T\) and as \(CP_j \leq \frac{b}{\alpha_j}\), it is deduced that \(\exists (d, Y) \geq T\). In this case, \(X_{P_j}\) is a (real) \((d, T, b)\)-MP candidate. As the capacity of any are not belonging to \(P_j\) is zero in \(X_{P_j}\), one sees that the capacity of \(P_j\) under any SSV, \(Y < X\), is less than \(\alpha_j\), and accordingly \(\exists (d, Y) > T\). It shows that \(X_{P_j}\) is a minimal candidate, and thus it is a (real) \((d, T, b)\)-MP according to Lemma 3. As a result, one can determine the corresponding \((d, T, b)\)-MPs to all the MPs which satisfy the budget constraint directly without generating any extra candidates. Now, we are at the point to state the proposed algorithm.

Algorithm 1 (assessing the reliability of smart grid communication network under time and budget constraints)

Input: A smart grid communication network \(G(N, A, M, L, C)\) with a demand level of \(d\), time limit of \(T\), budget limit of \(b\), and all the MPs, say \(P_1, P_2, \cdots, P_q\).

Output: The system reliability, that is, \(R(d, T, b)\).

Step 1. Calculate \(CP_j = \sum_{a_i \in P_j} c_r\) and \(LP_j = \sum_{a_i \in P_j} l_r\), for \(j = 1, 2, \cdots, q\). Let \(J = \{ j \mid CP_j \leq \frac{b}{\alpha_j} & LP_j < T \}\) and \(|J| = k\), that is, \(k\) is the number of MPs which satisfy the conditions.

Step 2. For each \(r = 1, 2, \cdots, k\), let \(j = J(r)\), that is, the \(r\)th index in \(J\), and \(\alpha_j = \left\lfloor \frac{d}{T - LP_j} \right\rfloor\). If \(\alpha_j \leq K P_j(M)\), then set \(X_{P_j} = (x_1, x_2, \cdots, x_m)\) as follows.

For \(i = 1, 2, \cdots, m, x_i = \begin{cases} \alpha_j & \text{if } a_i \in P_j, \\ 0 & \text{if } a_i \notin P_j. \end{cases}\)

Step 3. Assume that \(\sigma \leq k\) is the number of obtained \((d, T, b)\)-MP in Step 2. If \(\sigma = 0\), then \(R(d, T, b) = 0\), otherwise let \(E_j = \{ X \mid X \geq X_{P_j} \}\), for \(j = 1, 2, \cdots, \sigma\), and compute the system reliability by using the inclusion-exclusion technique through Eq. 5.

According to the discussions in this section and the preceding ones, the following theorem is at hand.

Theorem 2. Algorithm 1 evaluates the reliability of the given smart grid communication network correctly without generating any extra candidates.
IV. AN ILLUSTRATIVE EXAMPLE AND COMPLEXITY RESULTS

1) A descriptive example

Assume that the corresponding communication network in a smart grid is the one given in Fig. 3 with the provided arc data in Table I which includes the lead time, transmission cost, and probability distribution for the arcs’ capacities. The administrator needs to know the reliability of this communication network for transmitting \( d = 10 \) units of data from node 1 to node 5 within 8 units of time and budget of \( b = 50 \). We use Algorithm 1 for this aim. From Table I we have \( M = (5, 3, 4, 3, 2, 4, 5, 3) \), \( L = (2, 2, 3, 1, 2, 2, 3, 1) \) and \( C = (1, 2, 2, 1, 3, 2, 3, 4) \).

**Step 1.** We calculate \( CP = (CP_1, \ldots, CP_5) = (3, 5, 9, 5, 9, 6, 8, 8) \) and \( LP = (LP_1, \ldots, LP_5) = (4, 6, 6, 5, 5, 4, 8, 8) \). As \( \frac{b}{d} = 5 \) and \( T = 8 \), we have \( J = [1, 2, 4, 5] \) and \( k = 4 \).

**Step 2.** We calculate \( \alpha_1 = 3 \), \( \alpha_2 = 5 \), \( \alpha_3 = 4 \), \( \alpha_4 = 4 \), \( K_P = 4 \), \( K_P = 3 \), \( K_P = 3 \) and \( K_P = 3 \). As \( \alpha_j > K_P \), for \( j = 2, 4, 5 \), and then we set only \( X_P = (3, 0, 0, 0, 3, 0, 0) \) \( \sigma \).

**Step 3.** As \( \sigma = 1 \), we have \( E_1 = \{ X \mid X \geq (3, 0, 0, 0, 0, 0, 0) \} \), and accordingly \( R(d,T,b) = 0.68 \) which is not so good. For sure if the administrator increase the time or budget limits, the network reliability may increase.

This example shows how efficiently the algorithm removes several MPs to lessen the required work. One notes that finding all the \((d, T, b)\)-MP candidates and removing the non-minimal ones requires much more time than the consumed time in Algorithm 1. It is worthy to see that according to \( M = (5, 3, 4, 3, 2, 4, 5, 3) \), there are 172800 possible SSVs for this network, whereas Algorithm 1 directly found the only existing \((10, 8, 50)\)-MP.

2) Complexity results

We first recall that \( m \) and \( q \) are respectively the numbers of arcs and MPs. The time complexity of Step 1 to calculate \( CP_j \) and \( LP_j \), for \( j = 1, \ldots, q \), and remove the ones which satisfy the desired conditions is \( O(mq) \). As \( k \) is the number of MPs in Step 2, the time complexity of this step to compute \( \alpha_j \) and set the vectors, if any, is \( O(mk) \). Finally, as \( \sigma \) vectors arrive at Step 3, the time complexity of calculating the union probability in this step is of order of \( O(m\sigma^2) \). As \( \sigma^2 > q \) & \( k \), for large enough networks, the following theorem is at hand.

**Theorem 3.** The time complexity of Algorithm 1 to find all the \((d, T, b)\)-MPs is \( O(mq) \) and to assess the reliability of the network is \( O(m\sigma^2) \), where \( m \) and \( q \) are respectively the number of arcs and MPs in the network, and \( \sigma \) is the number of obtained \((d, T, b)\)-MPs in the algorithm.

V. EXPERIMENTAL RESULTS

To show the practical efficiency of our proposed algorithm in comparison with the available algorithms in the literature, we compare our algorithm and the one proposed by Lin [30]. It is noted that the algorithm of [30] serves for the cases without the budget limit, and hence we add a new step into the algorithm to check all the solutions for the budget constraint. For more convenience and readability, we state Lin’s algorithm with a newly added step here as Algorithm 2.

**Algorithm 2** (based on Lin’s algorithm [30] for assessing the reliability of MFNs under time and budget constraints)

**Input:** An MFN \( G(N, A, M, L, C) \) with a demand level of \( d \), time limit of \( T \), budget limit of \( b \), and all the MPs, say \( P_1, P_2, \ldots, P_q \).

**Output:** The system reliability, that is, \( R(d,T,b) \).

**Step 1.** For each \( j = 1, 2, \ldots, q \):

1.1. Find the minimum integer-valued number \( v_j \) such that \( LP_j \geq \left[ \frac{d}{\sigma_j} \right] \leq T \).

1.2. If \( v_j \leq K_P \), then set \( X_P = (x_1, x_2, \ldots, x_m) \) as follows.

For \( i = 1, 2, \ldots, m \), \( x_i = \left\{ \begin{array}{ll} \alpha_j & \text{if } a_i \in P_j, \\ 0 & \text{if } a_i \notin P_j. \end{array} \right. \)

**Step 2.** (Newly added step to check the solutions for the budget constraint) Remove every solution \( X_P \) that satisfies \( d \times CP_j \geq b \).

**Step 3.** Assuming \( \sigma \) as the number of obtained solutions, if \( \sigma > 0 \), let \( E_j = \{ X \mid X \geq X_P \} \), for \( j = 1, 2, \ldots, \sigma \), and compute \( R(d,T,b) = \Pr(\cup_{r=1}^{\sigma} E_r) \), else \( R(d,T,b) = 0 \).

As Step 3 in both algorithms 1 and 2 are to compute a union probability by using the available techniques in the literature, we compare the practicality of the algorithms based on their first two steps. Therefore, we use the algorithms to obtain all the SSVs under which \( d \) units of data can be sent from a source to a sink within a time of \( T \) and a budget of \( b \). This way, we compare the algorithms on the taken CPU seconds to solve the test problems.

To generate the numerical results, both algorithms are implemented in the MATLAB programming environment. We note that for implementing Algorithm 2, we use our presented results of Lemma 2 and accordingly the obtained results on this algorithm here are expected to be better than what could have been. We employ the Pan European topology, which is a rather large-size benchmark, depicted in Fig. 4 as well as one thousand large-size randomly generated test problems to compare the practical efficiency of the algorithms. All the numerical results were made on a computer with Intel(R) Core(TM) i5-2400S Duo CPU 3.1 GHz, with 8 GB of RAM.

For each network, the arcs’ capacities, lead times, and transmission costs take random integer values in the intervals \([10, 50], [5, 10], \) and \([5, 20] \), respectively. We note that with large enough limits of time and budget, any SSV can be a solution and there is nothing more to be solved by the algorithms. Hence, to have some meaningful limits, we considered \( T = \left\lfloor \frac{LP_1 + \cdots + LP_q}{q} \right\rfloor \) and \( b = \left\lfloor \frac{d(CP_1 + \cdots + CP_q)}{q} \right\rfloor \) as the time and budget limits in each test problem, respectively.

For the Pan European benchmark, we first calculate

\[ d^* = \left\lfloor \frac{K_P_1 + \cdots + K_P_q}{q} \right\rfloor, \]
Table I provides the arc data for Fig. 3.

| Arc Lead | Lead Cost | Capacity | Probability | Arc Lead | Lead Cost | Capacity | Probability |
|----------|-----------|----------|-------------|----------|-----------|----------|-------------|
| t         | 7         | 0.7      | 2           | 8.5      | 4         | 0.1      | 0.05        |
| a1       | 2         | 1        | 3           | 0.05     | 2         | 3        | 1           | 0.1         |
| a2       | 2         | 2        | 2           | 0.1      | 0         | 0.05     | 0.05        |
| a3       | 3         | 2        | 2           | 0.1      | 1         | 0.05     | 0.05        |
| a4       | 2         | 1        | 2           | 0.1      | 1         | 0.05     | 0.05        |

Table II provides the final results obtained on this network. The columns in this table are d, the demand level, Nh,A2, number of generated solutions in Algorithm 2 with transmission cost higher than b, Nr,A1, number of removed MPs with unaffordable transmission time or cost in the outset of Algorithm 1, Ns, number of solutions, t1, the running time of Algorithm 1, t2, the running time of Algorithm 2, and t2/t1 which is time ratio. The given CPU times in this table are in seconds.

| d | Nh,A2 | Nr,A1 | t1 | t2 | t2/t1 |
|---|------|-------|----|----|-------|
| 6 | 560  | 714   | 78 | 0.0170 | 0.0208 | 1.2237 |
| 7 | 560  | 714   | 78 | 0.0129 | 0.0190 | 1.4796 |
| 8 | 560  | 714   | 78 | 0.0127 | 0.0173 | 1.3624 |
| 9 | 560  | 714   | 78 | 0.0122 | 0.0175 | 1.4306 |
| 10| 560  | 714   | 78 | 0.0128 | 0.0176 | 1.3765 |
| 11| 551  | 714   | 78 | 0.0122 | 0.0167 | 1.3721 |
| 12| 550  | 714   | 78 | 0.0128 | 0.0165 | 1.3756 |
| 13| 550  | 714   | 78 | 0.0126 | 0.0166 | 1.3150 |
| 14| 550  | 714   | 78 | 0.0122 | 0.0165 | 1.3563 |
| 15| 549  | 714   | 78 | 0.0124 | 0.0165 | 1.3246 |

Geo. Mean 0.0130 0.0175 1.3471

Table II shows that Algorithm 1 removes 714 out of 1274 MPs in the outset for each case which is more than half of the MPs. Among these MPs, as seen from the fourth column, only 78 MPs are removed because of the high cost. Algorithm 1 solves all the cases (in average 1.34 times) faster than Algorithm 2 which shows the practical efficiency of our proposed algorithm.

To have a more meaningful comparison of the algorithms, we use one thousand randomly generated test problems. To this end, we consider n = 11, 12, · · · , 30 as the number of nodes and generate 50 random random networks for each case, making a total of 1000 test problems. To not have very complicated networks, we consider two integer numbers ℓ = 3(⌈n/2⌉−1) and γ = 25 − ⌈n/2⌉, where n is the number of nodes in the network. Then, the number of arcs in each corresponding generated network take random integer values.
Given the performance diagram in Fig. 5, one can see that around 10% of the test problems have been solved more than 50 times faster by Algorithm 1. Moreover, the horizontal axis shows that there exist some problems which have been solved by Algorithm 1 more than 400 times faster. In this profile, the algorithm whose performance diagram lies above the other is preferred [41].

All the numerical results including the tables II and III and Fig. 5 show the superiority of Algorithm 1 to Algorithm 2, and that as the network’s size grows this superiority increases.

VI. CONCLUSIONS

Communication networks play a crucial role in smart grids, and hence assessing the performance of such networks is of great importance. Moreover, the time and budget constraints are usually notable challenges in real-world systems. Therefore, taking into account both time and budget limits, to evaluate the performance of such networks, we considered a reliability index, which is the probability of transmitting $d$ units of data from a source to a destination through a single path within $T$ units of time and budget of $b$. Some results were presented based on which an efficient algorithm was proposed to address the problem. The algorithm was illustrated by using a benchmark network example. We also provided the complexity results. Moreover, several experimental results were provided by employing the Pan European topology and one thousand randomly generated networks. The numerical results clearly showed that our proposed algorithm outperforms some available algorithm in the literature, and that the superiority of our algorithm enhances as the network’s size grows.

ACKNOWLEDGMENTS

The second author thanks CNPq (grant 306940/2020-5) for supporting this work.

REFERENCES

[1] D. Bakken, Smart grids: clouds, communications, open source, and automation. CRC Press, 2014.
