Diameter-related properties of graphs and applications to network reliability theory

LOUIS PETINGI
College of Staten Island
City University of New York
Computer Science Department
2800 Victory Boulevard, Staten Island, N.Y.
USA
louis.petingi@csi.cuny.edu

Abstract: Given an undirected graph $G = (V, E)$, two distinguished vertices $s$ and $t$ of $G$, and a diameter bound $D$, a $D$-$s$, $t$-path is a path between $s$ and $t$ composed of at most $D$ edges. An edge $e$ is called $D$-irrelevant if does not belong to any $D$-$s$, $t$-path of $G$. In this paper we study the problem of efficiently detecting $D$-irrelevant edges and also study the computational complexity of diameter-related problems in graphs. Detection and subsequent deletion of $D$-irrelevant edges have been shown to be fundamental in reducing the computational effort to evaluate the Source-to-terminal Diameter-Constrained reliability of a graph $G$, $R_{\{s,t\}}(G, D)$, which is defined as the probability that at least a path between $s$ and $t$, with at most $D$ edges, survives after deletion of the failed edges (under the assumption that edges fail independently and nodes are perfectly reliable). Among other results, we present sufficient conditions to efficiently recognize irrelevant edges and we present computational results illustrating the importance of embedding a procedure to detect irrelevant edges based on these conditions, within the frame of an algorithm to calculate $R_{\{s,t\}}(G, D)$, built on a theorem of Moskowitz. These results yield a research path for the theoretical study of the problem of determining families of topologies in which $R_{\{s,t\}}(G, D)$ can be computed in polynomial time, as the general problem of evaluating this reliability measure is NP-Hard.

Key–Words: Network reliability, diameter constraint, paths, factoring, topological reductions.

1 Introduction

Unless otherwise stated, in this paper we consider undirected graphs $G = (V, E)$, where $V$ represents a finite set of vertices and $E$ is a finite set of edges.

The purpose of this work is two-fold. We first investigate, from a computational point of view, diameter properties of graphs related to the following optimization problem: given two vertices $s$ and $t$ of $G$, we would like to efficiently identify edges that do not belong to any path between $s$ and $t$ of length less or equal to a given bound $D$; we then apply some of the results shown to compute more efficiently the Diameter-Constrained reliability (DCR) of a communication network (originally introduced in [17]), a constrained version of the classical network reliability measure (refer to [20, 21, 22, 24, 25] for further discussion on this classical model). This study serves as a guide to address the problem of identifying families of topologies in which the DCR can be computed in polynomial time (as its computation is known to be NP-Hard).

Given a probabilistic graph $G = (V, E)$, a set of terminal nodes $K \subseteq V$, and a diameter bound $D$, in which each edge $e \in E$ has been assigned a probability of failure $q_e = 1 - r_e$ ($r_e$ is called the reliability of the edge $e$) under the assumption that edges fail independently and nodes are perfectly reliable, the $K$-terminal Diameter-Constrained reliability, $R_K(G, D)$, gives the probability of the event that for each pair of nodes $x, y \in K$, a path between $x$ and $y$ of length (i.e., number of edges comprising the path) $D$ or less, called a $D$-$x$, $y$-path, survives after deletion of the failed edges. In this paper we consider the case when $K = \{s, t\}$, known as the Source-to-terminal Diameter-Constrained reliability, denoted as $R_{\{s,t\}}(G, D)$. For the classical reliability measure, the $K$-terminal reliability $R_K(G)$, of a graph $G$, is the probability that after the removal of the failed edges, each pair of nodes $x, y \in K$ is connected by at least an operational path, independently of its length. Both the classical reliability and the DCR can be computed by application of a theorem of Moskowitz [14], also referred as the Factoring Theorem, in which the reliability of the probabilistic
graph is computed in term of the reliabilities of two smaller graphs using a specific edge as pivot.

In many real-life situations the quality of the communication depends on the existence of a path connecting each pair of terminals \( x \) and \( y \), whose length (measured as the number of edges) is bounded by a given integer \( D \). The \( K \)-terminal Diameter-Constrained Network reliability is the probability of this event and it was originally introduced by Petingi and Rodriguez in 2001 [17] (for a survey on this reliability model refer to [16]). The DCR can be applied to assess performance objectives of for example packet-oriented networks where links may fail and there is a "time-to-live" (TTL) limit, specified in number of hops that can be traversed by any given packet (for instance IPv6 packets include a hop limit field [11]). It is also the case of many overlay networks (such as peer-to-peer file sharing networks) that employ flooding protocols for peer discovery which specify a maximum number of hops to be visited by a request (for instance Gnutella, which employs a flooding-based routing algorithm with a TTL value of 7 hops [19]).

As the classical reliability measure does not capture the distances between the network nodes, the DCR can be applied to assess the probability of establishing a connection by setting, for example, the diameter bound \( D \) equal to the maximum number of hops to be visited by a packet or request. Another scenario for P2P networks is obtained if the link reliability value represents the probability that a pair of given nodes are in each other routing tables. In that case, the DCR models the fraction of the peers than can be reached from an arbitrary node. The DCR measure subsumes the classical reliability in the following sense: as any path in a network on \( n \) vertices is composed of at most \( n - 1 \) edges, then \( R_{K}(G, D) = R_{K}(G) \), whenever \( D = n - 1 \). As the calculation of the classical reliability for an arbitrary set of terminal vertices \( K \) is an NP-Hard problem [1], then evaluation of the DCR is an NP-Hard problem as well. For fixed number of terminal vertices \( K \), and for fixed values of the bound \( D \), Cancela and Petingi [7] proved that to determine \( R_{K}(G, D) \) is also an NP-Hard problem. Monte Carlo techniques have been shown to be excellent candidates to accurately estimate the classical reliability [4, 5] as well as to calculate the DCR [6].

For the classical reliability, topological reductions (e.g., series reduction, degree-2 reduction, parallel reduction, polygon-to-chain reductions) can be applied to a graph to reduce its size while preserving the reliability, and therefore reducing the computational effort for its evaluation. A series-parallel graph is a graph \( G = (V, E) \) that can be reduced to single edge by application of series-parallel reductions, and, for any arbitrary set of terminal vertices \( K \), \( R_{K}(G) \) can be computed in time \( O(|E|) \) [21]. For fixed values of the diameter bound \( D \), with the exception of parallel reductions, the aforementioned transformations cannot be applied to evaluate the DCR, as these topological transformations may change the distance between nodes of a graph (and therefore its diameter). As for this case these transformations are not reliability-preserving, the deletion of \( D \)-irrelevant edges can be considered as the alternative to efficiently calculate \( R_{K}(G, D) \), as well as possibly determine families of graphs in which \( R_{K}(G, D) \) can be calculated in polynomial time.

An edge \( e \) of a graph \( G \) is said to be irrelevant if deletion of the edge (denoted as \( G - e \)) preserves the \( K \)-terminal Diameter-Constrained reliability, that is, \( R_{K}(G, D) = R_{K}(G - e, D) \), for a given diameter bound \( D \). For the specific case when \( K = \{s, t\} \), an edge \( e \) that does not belong to any \( D-s, t \)-path can then be deleted without affecting the reliability. In [8] a preliminary study addressing \( D \)-irrelevancy was presented to show how the computational effort to evaluate the Source-to-terminal Diameter-Constrained reliability of graphs can be improved when irrelevant edges are efficiently detected (and then deleted); among other results we are extending this study by presenting new sufficient conditions to identify a superset of the \( D \)-irrelevant edges identified in [8].

The paper is structured as follows. In Section 2 we introduce notation and definitions pertaining to the concept of edge-irrelevancy and of the DCR. As the classical reliability is a special case of the Diameter Constrained reliability (i.e., when the diameter bound \( D = n - 1 \) for graphs on \( n \) vertices), in Section 3 we present a review of known reliability-preserving topological transformations applied to improve the computational effort to evaluate the classical reliability (and therefore of the DCR), and characterize families of graphs for which the reliability can be computed in polynomial time. Since for fixed values of \( D \), these transformations are not reliability-preserving, we introduce new sufficient conditions, in addition to the ones presented in [8], to efficiently identify irrelevant edges in graphs. In Section 4, we show how these sufficient conditions can be embedded within the framework of a procedure based on Moskowitz’s Theorem (i.e., Factoring Theorem) to evaluate \( R_{\{s,t\}}(G, D) \). In Section 5 we present a computational analysis of Factoring Theorem, when a procedure to detect irrelevant edges is present or not, in order to compute \( R_{\{s,t\}}(G, D) \), based on computational experiments (illustrated in Appendix A) performed on different families of topologies. Finally, in Section 6, we present conclusions and further research.
2 Definitions and notation

In this section we introduce definitions and notation that will be used in the sequel.

Definition 1 (Path) - An s, t-path is a sequence of distinct vertices < u_0 = s, u_1, u_2, \ldots, u_{r-1}, t = u_r > of a graph G, such that (u_i, u_{i+1}) is an edge of G, 0 \leq i \leq r - 1. If each edge e = (u, v) is assigned an integer weight \( w(e) \), the length of an s, t-path \( p =< u_0 = s, u_1, u_2, \ldots, u_{r-1}, t = u_r > \) is \( L_G(p) = \sum_{i=0}^{r-1} w((u_i, u_{i+1})) \). For the unweighted case, or equivalently when each edge is assigned a weight of one, then the length \( L_G(p) \) is the number of edges comprising the path \( p \).

Definition 2 (Simple Cycle) - A simple cycle is a sequence of vertices < u_0 = x, u_1, u_2, \ldots, u_{r-1}, y = u_r > of a graph G, such that (u_i, u_{i+1}) is an edge of G, 0 \leq i \leq r - 1, and all the vertices of the sequence are distinct except for \( x = y \).

Definition 3 (Irrelevancy and Critical graphs) - An s, t-path \( p \) is called a D-s, t-path if \( L(p) \leq D \) (i.e., the path length is at most D). Given a graph G = (V, E), a distinguished set of terminal vertices set K = {s, t}, and a diameter bound D, and edge e = (u, v) ∈ E is said to be D-relevant if e lies in some D-s, t-path, otherwise e is D-irrelevant. If every edge of G is D-relevant then G is called D-diameter-critical.

Definition 4 (Distance and Diameter) - The distance between two vertices x, y of G is distance_G(x, y) = min{L_G(p) : p is an x, y-path of G}. Moreover the K-diameter is the maximum distance between vertices of K.

Definition 5 (Degree) - The degree of a vertex v of a graph G, denoted as deg_G(v), is the number of edges incident at v.

We are also introducing notation to describe topological transformations in graphs.

○ Given two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), and two distinct vertices a, b such that \( a \in V_1 \) and \( b \in V_2 \), let \( G_1[a \cdot e \cdot b]G_2 \) describe the operation of joining \( G_1 \) and \( G_2 \) by an edge \( e = (a, b) \).

○ Given two paths \( p_1 =< u_0, u_1, u_2, \ldots, u_{r-1}, u_r > \), and \( p_2 =< v_0, v_1, v_2, \ldots, v_{m-1}, v_m > \), and two distinct vertices u and v of \( p_1 \) and \( p_2 \), respectively, let \( p_1[u \cdot e \cdot v]p_2 \) describe the operation of joining \( p_1 \) and \( p_2 \) by an edge \( e = (u, v) \).

In the following section we present a review of well-known reliability-preserving transformations applicable to the classical reliability for a graph \( G = (V, E) \) (and therefore applicable to evaluate the DCR when \( D = |V| - 1 \)) and the D-irrelevancy problem to efficiently identify irrelevant edges for fixed values of D.

3 Reliability-preserving transformations for the classical and Diameter Constrained reliability measures

3.1 Reductions applicable to the classical reliability and to the DCR when \( D = n - 1 \), for graphs on n vertices

This sub-section is intended as a brief review of topological reductions known to reduce the computational effort to evaluate the classical reliability (further detailed information can be found in [20, 21, 22, 24, 25]).

Given a probabilistic graph \( G = (V, E) \), the K-terminal Diameter Constrained reliability, \( R_K(G, D) \), is equal to the K-terminal reliability, \( R_K(G) \), whenever \( D = n - 1 \), where \( n = |V| \); consequently reliability-preserving reductions applicable for the classical case to reduce the computational effort to evaluate \( R_K(G) \), in addition to possibly identifying families of graphs in which the reliability can be evaluated in polynomial time (as the general problem is NP-Hard), can be extended for the evaluation of the DCR as well.

For fixed values of the diameter bound D, and for the case of the DCR, these reductions are not reliability-preserving, as they may change the distance between nodes of a graph in an arbitrary way; in the next sub-section we will look at D-irrelevancy as the alternative way to reduce the computation effort to evaluate \( R_K(G, D) \), for fixed D.

A simple reduction, is a reliability-preserving transformation that reduces the size of a graph \( G = (V, E) \), i.e., reduces \(|V| + |E|\), and therefore the computational complexity of evaluating \( R_K(G) \): Certain edges and/or vertices in G are replaced to obtain \( G' \), possibly a new set \( K' \) is defined; and a multiplicative factor \( \Omega \) is defined; all such that \( R_K(G) = \Omega R_K(G') \).

The following are simple reliability-preserving reductions [20]:

○ A parallel reduction replaces a pair of edges \( e_1 = (u, v) \) and \( e_2 = (u, v) \), with a single edge \( e_3 = (u, v) \) with reliability \( r_{e_3} = 1 - q_{e_1}q_{e_2} \), \( K' = K \), and \( \Omega = 1 \).
A series reduction replaces two edges \( e_1 = (u, v) \) and \( e_2 = (v, w) \), under the assumption that \( \text{deg}_G(v) = 2 \) (see Definition 5), with a single edge \( e_3 = (u, w) \) with reliability \( r_{e_3} = r_{e_1}r_{e_2} \), \( K' = K \), and \( \Omega = 1 \).

A degree-2 reduction replaces two edges \( e_1 = (u, v) \) and \( e_2 = (v, w) \), under the assumption that \( \text{deg}_G(v) = 2 \), and \( \{u, v, w\} \subseteq K \), with a single edge \( e_3 = (u, w) \) with reliability \( r_{e_3} = r_{e_1}r_{e_2}/(1 - q_{e_1}q_{e_2}) \), \( K' = K \setminus \{v\} \), and \( \Omega = 1 - q_{e_1}q_{e_2} \).

A graph \( G = (V, E) \) is called series-parallel if it can be reduced to a single edge by successive applications of series-parallel reductions mentioned above. If a probabilistic series-parallel graph \( G = (V, E) \), with terminal vertex set \( K \), can be reduced to a single edge, then this graph is called series-parallel-reducible and \( R_K(G) \) can be computed in polynomial time. However, depending upon the location of the terminal vertices, there are other series-parallel graphs that cannot be reduced to a single edge (series-parallel-irreducible) by successive applications of these simple reductions.

In [21], Satyanarayana and Wood introduced a new set of reliability-preserving non-simple topological transformations, called polygon-to-chain reductions, which in combination with the series-parallel and degree-2 reductions aforementioned, reduce any probabilistic series-parallel graph \( G = (V, E) \), and for any arbitrary set of terminal vertices \( K \subseteq V \), into a single edge in time \( O(|E|) \); consequently the evaluation of the classical reliability for any series-parallel graph is achievable in polynomial time.

A chain \( \xi \) of a graph \( G \) is an alternating sequence of vertices and edges \( u_0, (u_0, u_1), u_1, (u_1, u_2), u_2, \ldots, (u_{r-1}, u_r), u_r \), such that the internal vertices \( u_1, u_2, \ldots, u_{r-1} \) of the chain have all degree-2 in \( G \) and \( u_0 \), and \( u_r \) have degree greater than 2. The length of a chain is simply the number of edges it contains. Suppose next that \( \xi_1 \) and \( \xi_2 \) are two distinct chains of length \( l_1 \) and \( l_2 \) respectively; if the two chains have common end-vertices \( u \) and \( v \) in \( G \), then \( \Delta = \xi_1 \cup \xi_2 \) forms a polygon. Satyanarayana and Wood showed that any polygon, independently of its length, can be transformed, using the simple reductions, into seven elementary polygons in which terminal vertices were assigned to different vertices of these polygons. By a complex analysis, they showed that these elementary polygons could be always transformed into chains while preserving the calculation of the reliability.

For fixed values of the diameter bound \( D \), except for parallel reductions, the aforementioned reductions are not reliability-preserving in case of the DCR. In the next sub-section we consider how deletion of \( D \)-irrelevant edges could be considered as the alternative to minimize the computational effort to evaluate \( R_K(G, D) \), for constant values of \( D \) and when \( K = \{s, t\} \).

### 3.2 The problem of \( D \)-irrelevancy and efficient calculation of the Source-to-Terminal Diameter Constrained reliability

Given an edge \( e \) and two vertices \( s \) and \( t \) of a graph \( G \), \( e \) is \( D \)-relevant if and only if there exist an \( s, t \)-path \( p \) among all \( s, t \)-paths containing \( e \), such that \( L_G(p) \leq D \). Thus we are considering first the optimization problem of finding a shortest \( s, t \)-path of \( G \) containing an specific edge \( e \).

For the case in which negative integer weights can be assigned to the edges of the graph, we show next that to find a shortest \( s, t \)-path containing an specific edge \( e \) is NP-Hard, by transforming the Longest Path problem in which positive integer weights are assigned to the edges of a graph, into this decision problem:

**P1:** Shortest Path Containing an Specific Edge (SPE)

Instance: Graph \( G' = (V', E') \), edge \( e^* \), vertices \( s', t' \in V' \), length \( l(e') \in \mathbb{Z} \), for each \( e' \in E' \), and bound \( k' \in \mathbb{Z} \).

Question: Is there an \( s', t' \)-path \( p' \) of length \( k' \) or less containing the edge \( e^* \) (i.e., \( L(p') \leq k' \))?

Consider next the Longest Path problem, to be known to belong to the NP-Complete computational class, when positive integer weights are assigned to the edges [13]:

**P2:** Longest Path (SPP)

Instance: Graph \( G = (V, E) \), vertices \( s, t \in V \), length \( l(e) \in \mathbb{Z}^+ \), for each \( e \in E \), and bound \( k \in \mathbb{Z}^+ \).

Question: Is there an \( s, t \)-path \( p \) of length \( k \) or greater (i.e., \( L(p) \geq k \))?

**Lemma 6** The Shortest Path Containing an Specific Edge problem, P1, is NP-Hard.

**Proof:** It is not known if SPE is in NP. Consider the transformation from the Longest Path problem to the SPE problem.

Let \( G = (V, E) \), vertices \( s, t \in V \), length \( l(e) \in \mathbb{Z}^+ \), bound \( k \in \mathbb{Z}^+ \) be an instance of P2. Let \( G' = (V', E') \) be graph obtained by joining two copies of \( G \) by an edge \( e^* = (a, b) \) where \( a \) is vertex \( t \) of the first copy of \( G \) and \( b \) is vertex \( s \) of the second copy of \( G \),
that is, \( G' = Gt.e*.sG \). Moreover let the vertices \( s' \) and \( t' \) be the vertices of \( G' \) corresponding to vertex \( s \) of first copy of \( G \) and vertex \( t \) of the second copy of \( G \), respectively. In addition, for each edge \( e' \in E' \), let \( l(e') = -l(e) \) where \( e \) is \( e' \) corresponding edge in \( G \) (as the edges of \( G \) have been assigned positive weights, then the edges of \( G' \) have negative weights). Moreover let \( l(e^*) = -1 \), and \( k' = -2k - 1 \) (see Figure 1).

Suppose \( G \) contains an \( s,t \)-path \( p \) of length \( L_G(p) \geq k \). As \( L_{G'}(p') = -L_G(p) \) then the path \( p' = pt.e*.sp \) is an \( s',t' \)-path in \( G' \) containing \( e^* \). As \( L_{G'}(p') \geq k \) then \( L_{G'}(p') \leq -2k - 1 \).

Conversely let \( p' \) be an \( s',t' \)-path in \( G' \) of length \( L_{G'}(p') \leq k' \). The path \( p' \) must contain the edge \( e^* \), as otherwise \( p' \) is not a \( s',t' \)-path, then \( p' = p'_1t.e*.sp'_2 \), where \( p'_1 \) and \( p'_2 \) are \( s,t \)-paths in the first and second copy of \( G \), respectively. But as \( L_{G'}(p'_1) \leq k' = -(2k + 1) \) then either \( L_{G'}(p'_1) \leq -k \) or \( L_{G'}(p'_2) \leq -k \), then \( G \) contains a path whose length is greater or equal to \( k \), as \( p'_1 \) and \( p'_2 \) are paths in \( G' \).

If the optimization problem \( SPE \) is extended to find a shortest \( s,t \)-path containing a predefined set of edges (or vertices), then the problem is NP-Hard as well, as the above proof can be generalized by replacing the edge \( e^* \) by a path with a fixed number of edges, where each edge of the path is assigned a weight of -1.

When positive weights are assigned to the edges, little is known regarding the corresponding optimization problem of finding a shortest \( s,t \)-path containing an specific edge (or a predefined set of edges), even though this problem has been studied since the 1960s. In a recent paper by Björklund et al. [2], the authors studied the problem of finding a shortest simple cycle through a set of elements \( B \subset V \cup E \) (i.e., vertices and edges), and a randomized algorithm (of order \( 2|B|^{O(1)} \)) was introduced for any arbitrary set \( B \) of elements. This problem is equivalent to the problem of finding a shortest \( s,t \)-path through a set of edges (or vertices), as an edge between \( s \) and \( t \) can be added, if \( s \) and \( t \) are not adjacent, to form a simple cycle. However, according to this paper, little is known regarding the optimal solution of this optimization problem when \( b = |B| \) is a constant, \( b \geq 3 \). Thus it is not clear if there exist a computationally efficient way to determine necessary and sufficient conditions to classify an edge \( e \) as \( D \)-irrelevant or not. However in [8], sufficient conditions to identify \( D \)-irrelevant edges were introduced, and are stated in the following proposition:

**Proposition 7** Given a graph \( G = (V, E) \), a diameter bound \( D \in \mathbb{Z}^+ \), and an edge \( e = (u,v) \) of \( G \). If distance\(_G\)(s, u) + distance\(_G\)(v, t) ≥ D and distance\(_{G-e}\)(s, v) + distance\(_{G-e}\)(u, t) ≥ D, then \( e \) is \( D \)-irrelevant.

Given a graph \( G = (V, E) \), let \( n = |V| \) and \( m = |E| \). The distance (see Definition 4) between \( s \) and any other vertex of the graph, can be efficiently determined by application of Dijkstra’s Shortest Path algorithm (DSPA for short) [12] of order \( O(m + n) \). From Definition 3, it follows that edges that do not belong to any \( s,t \)-path are also \( D \)-irrelevant; it is important to note that if \( G \) is one-connected (i.e., there exist a vertex, called a cut-point, whose deletion disconnects \( G \)), and \( s \) and \( t \) belong to a two-connected component \( C = (V_c, E_c) \), any edge \( e \) that does not belong to this component, is also \( D \)-irrelevant as no \( s,t \)-path containing \( e \) exists in \( G \); thus it is possible that Proposition 7 won’t recognize \( e \) as irrelevant. In this case, edges that belong to blocks (two-connected components) other than \( C \), can be efficiently identified by a linear-time algorithm based on bi-connectivity theory [10]. Next suppose that \( G \) has more than one connected component, then if \( s \) and \( t \) belong to a same connected component (not necessarily a two-connected one) \( C = (V_c, E_c) \), any edge \( e \) that does not belong to \( C \) is also irrelevant. In this case Proposition 7 will classify this edge as \( D \)-irrelevant as the original distance between \( s \) and a end-point of \( e \) is set to infinte by Dijkstra’s algorithm.

We are proposing next sufficient conditions which detect a superset of the set of \( D \)-irrelevant edges identified by Proposition 7.

**Proposition 8** Given a graph \( G = (V, E) \), a diameter bound \( D \in \mathbb{Z}^+ \), and an edge \( e = (u,v) \) of \( G \). If distance\(_{G-e}\)(s, u) + distance\(_{G-e}\)(v, t) ≥ D and distance\(_{G-e}\)(s, v) + distance\(_{G-e}\)(u, t) ≥ D, then \( e \) is \( D \)-irrelevant.
Proof: Suppose that \( e = (u, v) \) is \( D \)-relevant, then as \( e \) belongs to a \( D \)-s, \( t \)-path, at least one of the following conditions is true:

1. There exist an \( s, t \)-path \( p = s = u_1, u_2, \ldots, \) \( u, v, \ldots, u_L, u_{L+1} = t \) > of length \( L \leq D \). Thus the length of sub-path \( p_1 = s = u_1, u_2, \ldots, u > \) plus the length of sub-path \( p_2 = v, \ldots, u_L, u_{L+1} = t > \) is at most \( D - 1 \).

2. There exist an \( s, t \)-path \( p = s = v_1, v_2, \ldots, v, u, \ldots, u_L, u_{L+1} = t \) > of length \( L' \leq D \). Thus the length of sub-path \( p_3 = s = v_1, v_2, \ldots, v > \) plus the length of sub-path \( p_4 = u, \ldots, u_L, u_{L+1} = t > \) is at most \( D - 1 \).

As the sub-paths \( p_1, p_2, p_3, \) and \( p_4 \) are paths in \( G - e \), then

- \( distance_{G-e}(s, u) + distance_{G-e}(v, t) < D \), or
- \( distance_{G-e}(s, v) + distance_{G-e}(u, t) < D \).

By the contrapositive, if \( (distance_{G-e}(s, u) + distance_{G-e}(v, t) \geq D) \) and \( (distance_{G-e}(s, v) + distance_{G-e}(u, t) \geq D) \) then \( e \) is \( D \)-irrelevant □

It is obvious that if the conditions stated in Proposition 7 recognize a set of \( D \)-irrelevant edges \( S_1 \), then the conditions specified by Proposition 8 find a set of \( D \)-irrelevant edges \( S_2 \) where \( S_1 \subseteq S_2 \).

In Figure 2 a graph is depicted in which edges \((1, 2), (2, 3), \) and \((3, 4)\) are \( D \)-irrelevant, whenever \( D = 5 \). In this case \( S_1 = \emptyset \) as the conditions stated by Proposition 7 are not met. However by application of Proposition 8, if \( e = (1, 2) \), \( (distance_{G-e}(s, 1) + distance_{G-e}(2, t) \geq 5) \) and \( (distance_{G-e}(s, 2) + distance_{G-e}(1, t) \geq 5) \), as \( distance_{G-e}(s, 1) = 1, distance_{G-e}(2, t) = 4, distance_{G-e}(s, 2) = 4, \) and \( distance_{G-e}(1, t) = 4 \). Similarly edges \((2, 3), \) and \((3, 4)\) are also identified as \( D \)-irrelevant by Proposition 8.

We next present two procedures, \textbf{Irrelevant-} \( P_1 (\) \((8)\)) and \textbf{Irrelevant-} \( P_2 (\), to detect irrelevant edges, based on Proposition 7 and Proposition 8, respectively:

\textbf{Irrelevant-} \( P_1 (\)

\textit{Input:} Graph \( G = (V, E) \), terminal set \( K = \{s, t\} \), diameter bound \( D \).

1. delete from \( G \) edges that do not belong to any \( s, t \)-path.

2. call \textbf{DSPA} twice, once to find the \( distance_G(s, v) \) for every vertex \( v \in V \), and then to find \( distance_G(t, v) \) for every vertex \( v \in V \) (order \( O(m + n) \)).

3. delete the edges from \( G \) that meet the conditions stated in Proposition 7.

4. if there still edges in \( G \) that do not belong to any \( s, t \)-path, go to Step 1, else Exit.

\textbf{Irrelevant-} \( P_2 (\)

\textit{Input:} Graph \( G = (V, E) \), terminal set \( K = \{s, t\} \), diameter bound \( D \).

1. delete from \( G \) edges that do not belong to any \( s, t \)-path.

2. for every edge \( e = (u, v) \) of \( G \) do

2.1 delete \( e \) from \( G \). Apply \textbf{DSPA} to \( G - e \).

2.2 if conditions stated in Proposition 8 for the end-vertices of \( e = (u, v) \) are met Skip (the edge is \( D \)-irrelevant thus consider \( G - e \)), else put-back \( e \) in \( G \).

3. if there still edges in \( G \) that do not belong to any \( s, t \)-path, go to Step 1, else Exit.

In Step 1 of procedures \textbf{Irrelevant-} \( P_1 (\) and \textbf{Irrelevant-} \( P_2 (\), all the edges that do not belong to any \( s, t \)-path can be identified by a linear-time algorithm based on bi-connectivity theory [10].

Despite of the fact that procedure \textbf{Irrelevant-} \( P_2 (\) finds a superset of the edges detected by \textbf{Irrelevant-} \( P_1 (\), the conditions stated in Step 2 of the latest are determined by just two application of Dijkstra’s algorithm in time \( O(m + n) \), while the distance conditions stated in Proposition 8 are determined in time \( O(m^2) \), as we must apply \textbf{DSPA} \( m \) times (i.e. Step 2 of \textbf{Irrelevant-} \( P_2 (\)), each time when we delete a possible irrelevant edge \( e \) from \( G \); the trade-off between the number of irrelevant edges recognized, and the computational complexity for detecting these edges, when applying these procedures, has to be further investigated (Section 5).

As it was mentioned in the Introduction, both the classical reliability and the \( DCR \) can be computed by application of a procedure based on a theorem of Moskowitz (i.e., Factoring Theorem), in which the reliability of the probabilistic graph \( G \) can be computed in term of the reliabilities of two (possibly smaller) networks derived from \( G \) by fixing the state of a selected edge \( e \) either up (i.e., \( e \) is operational) or down (i.e., \( e \) failed). In the next section we consider the problem of embedding a procedure to determine irrelevant edges (by either applying \textbf{Irrelevant-} \( P_1 (\) or \textbf{Irrelevant-} \( P_2 (\)) within the context of Moskowitz’s Theorem.
4 Moskowitz and computation of the Source-to-terminal Diameter-Constrained Reliability

Moskowitz’s Decomposition Theorem express the reliability of a network $G$ as a function of the reliabilities of the two networks obtained from $G$ by fixing the state of a selected edge $e$ either up (i.e., $r_e$ is set to 1) or down (i.e., $r_e$ is set to 0). Moskowitz’s decomposition was extensively used within the context of the classical reliability (see [15, 18, 22, 24, 25]). Within this context we say that the random state of an edge $e$ is undetermined if $0 < r_e < 1$ [8].

Theorem 9 For any network $G$ that has at least one edge $e$ whose random state is undetermined then

$$R_{\{s,t\}}(G, D) = r_e R_{\{s,t\}}(G \ast e, D) + (1 - r_e) R_{\{s,t\}}(G - e, D),$$

where

- $e$ is an edge with undetermined random state in $G$ if $0 < r_e < 1$.
- $G \ast e$ is the network obtained from $G$ by fixing the edge $e$ up (i.e., $r_e = 1$).
- $G - e$ is the network obtained from $G$ by fixing edge $e$ down (i.e., $r_e = 0$, or equivalently $e$ is deleted from $G$).

Consider a procedure Factoring( ) to evaluate $R_{\{s,t\}}(G, D)$, derived from Theorem 9; this procedure describes a binary tree in which each node $j$ of this tree represents a subgraph of $G$, $G_j$, in which its edges are either operational (reliability 1), have failed (reliability 0), or whose random states are undetermined (the root node of the derived binary tree represents the original network $G$). For each of the possible subgraphs $G_j$’s, its Source-to-terminal Diameter-

Constrained reliability is then calculated as:

$$R_{\{s,t\}}(G_j, D) = \begin{cases} 
0 : & \text{if there is no } D-s, t \text{-path in } G_j. \\
1 : & \text{if } G_j \text{ contains an operational } D-s, t \text{-path.} \\
r_e R_{\{s,t\}}(G_j \ast e, D) + (1 - r_e) R_{\{s,t\}}(G_j - e, D) : & \text{if } e \text{ is undetermined.}
\end{cases}$$

We can embed either procedure Irrelevant-$P_1()$ or procedure Irrelevant-$P_2()$, described in Section 3, within Factoring( ), to possibly delete irrelevant edges in each of the states $G_j$ of the binary tree generated by the application of the recursive function previously stated, to possibly shorten the computational effort. The following procedure, called Fact-Reductions$-i()$, evaluates the Source-to-terminal Diameter-Constrained reliability of a network while enforcing the deletion of irrelevant edges identified by procedure Irrelevant-$P_i()$ ($i \in \{1, 2\}$). Procedure Fact-Reductions$-i()$ receives five parameters, namely the network topology $G$, the source and terminal nodes $s$ and $t$, the diameter constraint $D$, and a flag called flagr, which indicates whether further reductions are or are not possible. At the first invocation, flagr is set to 1.

Procedure Fact-Reduction$-i(G, s, t, D, flagr)$

Input: network $G = (V, E)$, $s, t, D$, and flagr

Output: reliability $R_{\{s,t\}}(G, D)$

1. Check end recursion condition:

   1.1. If $G$ contains a $D-s, t$-path having only operational edges return (1).
   1.2. If there is no $D-s, t$-path in $G$ return (0).

2. Apply procedure to detect irrelevant edges:

   2.1. If (flagr = 1) call Irrelevant-$P_1(G, s, t, D)$.

3. Select randomly an edge $e$ in $G$ with undetermined state.

4. Solve recursively for $G - e : R_{\{s,t\}}(G - e, D) = $ Fact-Reduction$-i(G - e, s, t, 1)$.

5. Solve recursively for $G \ast e : R_{\{s,t\}}(G \ast e, D) = $ Fact-Reduction$-i(G \ast e, s, t, 0)$.

6. Compute $R_{\{s,t\}}(G, D) :$ return $(R_{\{s,t\}}(G, D) = (1 - r_e) R_{\{s,t\}}(G - e, D) + r_e R_{\{s,t\}}(G \ast e, D))$. 

Figure 2: Graph $G = (V, E)$ and diameter bound $D = 5$. 
In the next section we present experimental results to compare the computational performances of \texttt{Factoring()}, \texttt{Fact-Reduction$-1()$}, and \texttt{Fact-Reduction$-2()$}.

5 Analysis of experimental results illustrated in Appendix A

In this section we compare the computational effort and other performance metrics yielded by the three methods mentioned in Section 4, that is, \texttt{Factoring()}, \texttt{Fact-Reduction$-1()$}, and \texttt{Fact-Reduction$-2()$}, based on experimental results illustrated in Appendix A. Tests were performed on different families of topologies, which encompass the Arpanet, the 5X5-Grid, the Dodecahedron, and Circulant graphs $C^n_{1,n/2}$ on $n$ vertices with jumps 1 and $n/2$ (see Figure 3), and they were chosen on the basis of their previous use in the related literature. As most of these topologies are of relatively low-density, we have also performed tests on high-density graphs such as complete graphs ($K_n$) on $n$ vertices.

In [8] computational experiments were performed on the same families of topologies as the one shown in Figure 3, to compare the performance of \texttt{Factoring()}, with the performance of \texttt{Fact-Reduction$-1()$}, and by taking into account different values of the diameter bound $D$. The tests performed in this work were geared to complement the study presented in [8], by also presenting computational results obtained by embedding procedure \texttt{Irrelevant-$P_2()$} within \texttt{Factoring()}, and run on the same set of topologies.

For all tests, an unique probability of failure $q_e = 0.5$ was assigned to each edge $e \in E$. In Table 1 and Table 2 of Appendix A, the data shown in columns 1 through 3 represent the type of topology, the label of the source and terminal nodes of the topology, and the diameter bound $D$, respectively. Column 4 shows the value of the Source-to-terminal Diameter Constrained reliability; columns 5 and 6, 7 and 8, and 9 and 10, represent the tree size (i.e., the number of nodes generated by factoring recursive algorithm) and CPU time (in seconds) taken by \texttt{Factoring()}, \texttt{Fact-Reduction$-1()$}, and \texttt{Fact-Reduction$-2()$} procedures, respectively.

In agreement with the conclusions stated in [8], the experimental results illustrated by Table 1 and Table 2 show a consistent computational gain observed when elimination of irrelevant edges was performed on most topologies, either by applying \texttt{Fact-Reduction$-1()$} or \texttt{Fact-Reduction$-2()$}, especially when low-density topologies were tested (e.g., Circulants, Dodecahedron); the most significant gain was observed when tests were performed on the 5X5-Grid topology (Table 1). We think that detection of irrelevant edges may play an important role when studying the computational complexity for evaluating $R_{\{s,t\}}(G, D)$ for low-density graphs.

For topologies composed of $n$ nodes, the computational gain when detecting irrelevant edges is particularly important for low values of the diameter bound $D$, and it becomes less significant when $D$ increases toward the maximum value $n - 1$ (i.e., as stated in the Introduction $R_{\{s,t\}}(G, D)$ approaches the classical reliability value $R(s,t)(G)$), or when tests were performed on high-density topologies (e.g., $K_7$ or $K_8$).

The comparison between \texttt{Fact-Reduction$-1()$} and \texttt{Fact-Reduction$-2()$} yields a computational gain when the latest was applied on most of the classes of topologies, and for most values of the diameter bound $D$, except when the $D$ bound was approaching $n - 1$, or when tests were conducted on high-density graphs. It is important to note that the number of recursive calls (i.e., tree size) was consistently less when \texttt{Fact-Reduction$-2()$} was applied on most topologies.

6 Conclusions and further research

The purpose of this work is to present a study of the optimization problem of identifying $D$-irrelevent edges and diameter-related problems in graphs, from a computational point of view. Since presently is unknown if necessary and sufficient conditions that can efficiently (polynomial time) detect all irrelevant edges exist (the equivalent problem when edges with negative integer weights are allowed is NP-Hard), we have introduced new sufficient conditions that can efficiently recognize a subset of the $D$-irrelevant edges. We then presented numerical results illustrating the importance of embedding a procedure to detect irrelevant edges based on these conditions, in combination with an algorithm derived from Moskowitz’s Decomposition Theorem, in order to evaluate $R_{\{s,t\}}(G, D)$.

Future work will comprise the determination of classes of topologies for which $R_{\{s,t\}}(G, D)$ can be evaluated in polynomial time (e.g., sparse graphs) as suggested by the computational analysis presented in Section 5.

Acknowledgements: The research was partially supported by PSC-CUNY Research Grant: 64588-00-42 from the City University of New York Research Foundation.

References:


7 Appendix A (experimental results)

Figure 3: Classes of topologies: a) 5X5-Grid, b) Arpanet, c) Dodecahedron, and d) Circulant $C_{16}^{16}$.

<table>
<thead>
<tr>
<th>$G$</th>
<th>$(s, t)$</th>
<th>$D$</th>
<th>$R_{st}(G, D)$</th>
<th>Fact</th>
<th>Fact-R-1</th>
<th>Fact-R-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5X5Grid</td>
<td>(0,20)</td>
<td>8</td>
<td>0.176359</td>
<td>N/A</td>
<td>&gt; 24 hours</td>
<td>317631</td>
</tr>
<tr>
<td>5X5Grid</td>
<td>(0,6)</td>
<td>8</td>
<td>0.498971</td>
<td>353362807</td>
<td>883</td>
<td>43661</td>
</tr>
<tr>
<td>5X5Grid</td>
<td>(0,12)</td>
<td>8</td>
<td>0.339515</td>
<td>1.09e +09</td>
<td>2706</td>
<td>16333687</td>
</tr>
<tr>
<td>5X5Grid</td>
<td>(0,18)</td>
<td>8</td>
<td>0.234521</td>
<td>N/A</td>
<td>&gt; 24 hours</td>
<td>1.005e +09</td>
</tr>
<tr>
<td>5X5Grid</td>
<td>(0,24)</td>
<td>8</td>
<td>0.123324</td>
<td>N/A</td>
<td>&gt; 24 hours</td>
<td>1.035e +09</td>
</tr>
</tbody>
</table>

Table 1: Comparison of the three methods (Factoring, Fact-Reduction–1(), and Fact-Reduction–2()) on the 5X5-Grid. Computations whose execution time exceeded 24 hours were aborted.
<table>
<thead>
<tr>
<th>$G$</th>
<th>$(s, t)$</th>
<th>$D$</th>
<th>$R_{s,t}(G, D)$</th>
<th>tree-size</th>
<th>CPU-t (s)</th>
<th>Fact-R-1</th>
<th>tree-size</th>
<th>CPU-t (s)</th>
<th>Fact-R-2</th>
<th>tree-size</th>
<th>CPU-t (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dodeca</td>
<td>(0,19)</td>
<td>5</td>
<td>0.168441</td>
<td>44015523</td>
<td>95</td>
<td>27935</td>
<td>&lt;1</td>
<td>27935</td>
<td>&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>0.268820</td>
<td>79223841</td>
<td>170</td>
<td>19313039</td>
<td>89</td>
<td>11353599</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>0.285391</td>
<td>92836783</td>
<td>201</td>
<td>66597743</td>
<td>298</td>
<td>31235671</td>
<td>255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arpanet</td>
<td>(0,19)</td>
<td>4</td>
<td>0.162109</td>
<td>13803603</td>
<td>19</td>
<td>97</td>
<td>&lt;1</td>
<td>97</td>
<td>&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.237618</td>
<td>62130037</td>
<td>103</td>
<td>362293</td>
<td>1</td>
<td>80109</td>
<td>&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>0.295711</td>
<td>214988177</td>
<td>373</td>
<td>106920719</td>
<td>399</td>
<td>61103239</td>
<td>371</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>0.302415</td>
<td>263588889</td>
<td>475</td>
<td>261395215</td>
<td>876</td>
<td>109031949</td>
<td>690</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_6$ (0,5)</td>
<td>3</td>
<td>0.913330</td>
<td>44673</td>
<td>&lt;1</td>
<td>32305</td>
<td>&lt;1</td>
<td>29731</td>
<td>&lt;1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.923584</td>
<td>45417</td>
<td>&lt;1</td>
<td>45369</td>
<td>&lt;1</td>
<td>42655</td>
<td>&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_7$ (0,6)</td>
<td>3</td>
<td>0.953217</td>
<td>2634387</td>
<td>9</td>
<td>1535477</td>
<td>22</td>
<td>1454955</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.962086</td>
<td>2675467</td>
<td>8</td>
<td>2593747</td>
<td>40</td>
<td>2482975</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.963001</td>
<td>2679667</td>
<td>8</td>
<td>2673787</td>
<td>38</td>
<td>2566875</td>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_8$ (0,7)</td>
<td>2</td>
<td>0.911011</td>
<td>277526209</td>
<td>1550</td>
<td>2915</td>
<td>0</td>
<td>2915</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>0.975735</td>
<td>313904521</td>
<td>1185</td>
<td>140021817</td>
<td>1143</td>
<td>135421811</td>
<td>1343</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.982520</td>
<td>317848021</td>
<td>1197</td>
<td>317225461</td>
<td>2557</td>
<td>308845439</td>
<td>3358</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>0.982573</td>
<td>317878981</td>
<td>234</td>
<td>317877541</td>
<td>546</td>
<td>309523559</td>
<td>678</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{20}^{1,10}$ (0,19)</td>
<td>5</td>
<td>0.578125</td>
<td>207057</td>
<td>0</td>
<td>337</td>
<td>0</td>
<td>157</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>0.581543</td>
<td>1592451</td>
<td>2</td>
<td>7013</td>
<td>0</td>
<td>2813</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>0.584226</td>
<td>6190747</td>
<td>11</td>
<td>1264923</td>
<td>4</td>
<td>1185547</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>17</td>
<td>0.587613</td>
<td>15617515</td>
<td>27</td>
<td>15188151</td>
<td>48</td>
<td>6622651</td>
<td>33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Comparison of the three methods ([Factoring()](#), [Fact-Reduction−1()](#), and [Fact-Reduction−2()](#)) on the Dodecahedron, Arpanet, Complete graphs on 6, 7, 8 vertices, and the Circulant on 20 vertices with jumps 1 and 10.