On the complexity of the Diameter-Constrained K-Reliability Problem

Abstract: Consider a communication network with perfect nodes whose links fail independently, a given subset $K$ of distinguished terminals, and a natural diameter $d$. The corresponding $d$-diameter constrained $K$-reliability ($d$-DCKR) is the probability that the $K$ terminals remain connected, with at least one path with length not higher than $D$. This problem has valuable applications in hop-constrained communication, for instance, in degraded channels, flooding-based systems and peer-to-peer networks.

The general $d$-DCKR is inside the class of NP-Hard computational problems, since the connectivity of a random network with imperfect links is a particular subproblem of it. In this article, we prove that the computational complexity of the 2-DCKR is linear in the number of sites of the network for any $|K|$ fixed, and we give an analytic expression for the target probability in terms of the probability of operation $p(xy)$ for each link $xy$ of the network.

We introduce two Monte Carlo-based heuristics to tackle the general $d$-DCKR problem. The first one connects the target problem with polynomial interpolation theory, whereas the second counts subgraphs with Crude-Monte Carlo. The effectiveness of both approaches is illustrated on the lights of Petersen and Dodecahedron graphs. The article concludes with a discussion of trends for future work.

Keywords: Network reliability, survivability, computational complexity.

Reference to this paper should be made as follows: <List of authors> (xxxx) ‘On the complexity of the Diameter-Constrained K-Reliability Problem’, Int. Journal of Metaheuristics, Vol. x, No. x, pp.xxx–xxx.

1 Introduction

Consider a communication network where sites are perfect but links can fail independently. Every link $xy$ is either operational or not, with respective probabilities denoted by $p(xy)$ and $1 - p(xy)$. Therefore, there is an “operational subnetwork” composed by all the sites, but those links that are operational. The classical $K$-reliability problem aims to find the probability that a given subset $K$ of distinguished sites remains connected, and it has been widely studied by Colbourn (1987). Our goal is to understand a related problem, known as $d$-diameter-constrained $K$-reliability ($d$-DCKR), where distinguished sites must be connected by not more than $d$ hops, and the reliability is denoted by $R_K(G, d)$. This problem is inspired in degraded systems (i.e. electrical networks, noisy telecommunication...
and latency-sensitive applications over the Internet (i.e. live video streaming distribution in Content Delivery Networks, VoIP, among others). The classical $K$-terminal reliability is in the class of NP-Hard problems Provan and Ball (1983). An immediate corollary is that the general $d$-DCKR problem is NP-Hard, since it is a generalization (the classical problem occurs when $d > |V| - 1$).

The authors Cancela and Petingi (2001) show that the 2-DCKR can be solved in polytime when $|K| = 2$. Indeed, given two terminals $s$ and $t$ we can explicitly determine all paths with length two or less between them linearly in the number of nodes, as well as their probabilities of operation. However, they also proved that when the diameter $d$ is fixed in an arbitrary value $d \geq 3$, the $d$-DCKR is NP-Hard even when $|K| = 2$.

The main result of this article is that the 2-DCKR is in the computational class $P$ of polynomially solvable problems, for any fixed value for $|K|$. Moreover, its computational complexity is linear in the number of sites $|K|$. A complete proof is provided in Section 2.

Additionally, we introduce two heuristics in order to address the general $d$-DCKR problem. Specifically, a naive heuristic called $F$-MonteCarlo and a more sophisticated one called $d$-Interpol are introduced in Section 3. The former is strictly connected with the traditional MonteCarlo machinery as an approximation technique, whereas the latter connects the target problem with classical polynomial interpolation theory. Both heuristics are tested with the so-called Petersen and Dodecahedron graphs, in Section 4. Interestingly enough, $F$-MonteCarlo though simple performs better than the novel sophisticated technique, which results numerically unstable. Nevertheless, the latter can suffer improvements inspired in interpolation theory and rare event simulation, which are discussed in Section 5. Finally, concluding remarks and open problems are detailed in Section 6.

2 The 2-DCKR is in $P$

In this section we prove the 2-DCKR is in the class $P$ of polynomially-solvable algorithms. What is more:

**Theorem 1** The computational complexity of the 2-DCKR is linear in the number of sites provided that $|K|$ is fixed.

The proof is organized as follows. A bit of elementary terminology and sketch of the proof is provided in Subsections 2.1 and 2.2. A partition of cases and their probabilities are obtained in Subsections 2.3 and 2.4. The proof is closed in Subsection 2.5.

2.1 Terminology

Let us model the network by a simple, undirected and complete graph $G = (V, E)$, with $n = |V|$ and $E = \{(x, y) : x \in V \land y \in V \land x \neq y\}$. The nodes $V$ correspond to the sites of the network. Each edge $xy \in E$ has a label $p(xy) \in [0,1]$. If there is a link connecting the sites to which $x$ and $y$ correspond then $p(xy)$ is the probability that it is operational; otherwise we define $p(xy) = 0$. We denote the probability of any event $z$ as $\Pr(z)$. Additionally:
given \( K \subseteq E, \) a 2-path is any \( E' \subseteq E \) such that the partial graph \( (V, E') \) is 2-\( K \)-connected;

- we denote by \( n' \) the number of nodes not in \( K \), that is, \( n' = n - |K| \);

- we denote by \( O^2_{E}(K) \) the set of 2-paths determined by \( E \) and \( K \) (following the notation of Petingi (2008));

- we denote by \( X^{(m)} \) the set of all subsets of a certain set \( X \) that have \( m \) different elements, that is, \( X^{(m)} = \{ Y \subseteq X : |Y| = m \} \);

- we denote by \( \otimes \) the binary operator defined as:
  \[
  (a_1, a_2, \ldots, a_n) \otimes (b_1, b_2, \ldots, b_n) = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\};
  \]

- we call requirement to any set of two nodes \( \{x, y\} \subseteq K^{(2)} \) and denote it by \( xy \). We say that a requirement is satisfied if there is a path of length below three, formed only by operating edges, that connects the two nodes that define the requirement;

- we denote by \( \mathcal{P}(A) \) the powerset of a certain set \( A \).

### 2.2 Sketch of the Proof

We start by finding an analytical expression for computing \( R_K(G, 2) \). To do so, we partition \( O^2_{E}(K) \) in disjoint components whose probabilities are then computed and totaled. We build a function \( f : O^2_{E}(K) \to A \) (with a certain discrete codomain \( A \) conveniently chosen) and then compute the probability of the domain by totaling the probabilities of all preimages of \( A \):

\[
R_K(G, 2) = \Pr(O^2_{E}(K)) = \sum_{a \in A} \Pr(f^{-1}(a)) \quad (1)
\]

The set \( A \) is defined as

\[
A = \bigcup_{\ell=0}^{n'} \left( (V \setminus K)^{(\ell)} \otimes \mathcal{P}(K)^{(\ell)} \right).
\]

Each element of \( A \) is a set of pairs \( (t, C) \), where \( t \) is a node of \( V \setminus K \) and \( C \) is a subset of \( K \). We see each such element of \( A \) as a collection of sets of edges between nodes of \( V \setminus K \) and \( K \) that belong to a 2-path (besides edges between nodes of \( K \)). Observe that any edge linking two nodes of \( V \setminus K \) is irrelevant for the 2-\( K \)-connectivity, since they can not be part of a path of length one or two connecting two nodes of \( K \). The function \( f \) is defined later in Eq. (2). We first show that \( A \) can be built such that totaling the probabilities \( \Pr(f^{-1}(a)) \) involves a number of elementary operations that is polynomial in \( n \); finally, we show that it is indeed linear in \( n \). A more detailed version of the proof herein presented can be seen in Canale et al. (2013).
2.3 Partitioning $O^2_K(E)$

We assume that there is a certain strict ordering within $V$. We say that a family $C \subseteq \mathcal{P}(K)$ covers (or is a cover of) $K$ for $F \subseteq K^{(2)}$, and denote it by $C \supseteq F K$, if and only if for every requirement $xy$ at least one of the following applies: (i) $xy \in F$; (ii) $\exists z \in K : \{xz,zy\} \subseteq F$; (iii) $\exists C \in \mathcal{C} : \{x,y\} \subseteq C$.

![Figure 1 Example of cover set](image)

Figure 1 illustrates the concept of covers. The thick lines represent the elements of $F$. The nodes connected by thin lines to the square nodes represent each element of $C = \{C_1, C_2, C_3\}$. Condition (i) applies for the pairs $ab, ac$ and $cd$. Condition (ii) applies for $bc$ and $ad$. Condition (iii) applies for $ac, cd, bc, bd$ and $cd$. Thus all pairs satisfy at least one of (i), (ii), (iii) and then $C \supseteq F K$.

Observe that if $E' \in O^2_K(E)$ then the family $C_{E'}$ defined as the set

$$C_{E'} = \{C_{E'}(t) : t \in V \setminus K\} \text{ being } C_{E'}(t) = \{x \in K : xt \in E\},$$

covers $K$ for $F = E' \cap K^{(2)}$, and we say that $E'$ generates $C_{E'}$. Conversely, given a cover $C$ of $K$ for $F$, if $|C| \leq n - |K|$ then there is some $E'$ that generates $C$; for example $E' = \{xy \in E : xy \in F \vee (\exists C \in \mathcal{C} : \{x,y\} \subseteq C)\}$ is a 2-path. The preceding definitions relate with the operational states of the network in the following way. Assume that the elements of $F$ correspond exactly to the edges linking nodes of $K$ that operate at a certain moment. Assume also that each element $C \in \mathcal{C}$ represents a node in $V \setminus K$ whose neighbors, ignoring failing edges, are exactly the elements of $C$ (e.g. $C_1, C_2, C_3$ in Figure 1). Then, $C \supseteq F K$ implies that every requirement is satisfied at that moment. The powerset $\mathcal{P}(\mathcal{K})$ has $2^{|K|}$ elements (i.e. there are $2^{|K|}$ different subsets of $K$). Its powerset $\mathcal{P}(\mathcal{P}(\mathcal{K}))$ has $2^{|2^{|K|}|}$ elements that are sets of subsets of $K$. So, for the number of families $\mathcal{C}$ for which $C \supseteq F K$, there is an upper bound $2^{2^{|K|}}$ that depends only on $|K|$. Now, observe that when considering a family $C_{E'}$, if there are two nodes $t \neq t' \notin K$ with $C_{E'}(t) = C_{E'}(t')$, one of $t$ and $t'$ can be removed from the graph $G$, yet obtaining the same family $C_{E'}$. For example, in Figure 1, the addition of a node $C_4$ with exactly $a$ and $c$ as neighbors in $K$, would make no difference when producing $C_{E'}$, due to the existence of a node $C_1$ with exactly the same neighbours in $K$. In general, given a path $E'$ and one element $C$ of the cover $C_{E'}$ there are one or more $t \in V \setminus K$ such that $C_{E'}(t) = C$. Let us define $t_{E'}(C)$ as the minimum of them according to the ordering of $V$, that is

$$t_{E'}(C) = \min \{t \in V \setminus K : C_{E'}(t) = C\}.$$
Then, for every 2-path \( E' \), there is exactly one set \( \{t_1, t_2, \ldots, t_\ell\} \) and one cover \( \mathcal{C}_{E'} = \{C_1, C_2, \ldots, C_\ell\} \) for \( F = E' \cap K^{(2)} \), such that \( t_i = t_E(C_i) \) \( \forall i = 1, \ldots, \ell \).

Now we can define our function \( f \) as follows:

\[
f(E') = \{(t_1, C_1), (t_2, C_2), \ldots, (t_\ell, C_\ell)\}. \tag{2}
\]

### 2.4 Finding the probabilities

Given \( x \in V \setminus K \) and \( C \subseteq K \), let us denote as \( P(x, C) \) the event where the set of neighbors of node \( x \) that belong to \( K \) and are connected by working links is exactly \( C \). Its probability, denoted as \( p(x, C) \), is:

\[
p(x, C) = \prod_{y \in C} p(xy) \prod_{y \not\in K \setminus C} (1 - p(xy)).
\]

Now, given \( C_1, \ldots, C_\ell \subseteq K \) and \( t_1, \ldots, t_\ell \in V \setminus K \) such that \( t_i < t_{i+1} \) \( \forall i = 1, \ldots, \ell - 1 \), let us define the event \( P(t_1, \ldots, t_\ell, C_1, \ldots, C_\ell) \) as the event where the following statements hold true:

- the nodes \( t_1, \ldots, t_\ell \) are connected exactly to \( C_1, \ldots, C_\ell \) in \( K \) respectively;
- the nodes \( t \) with \( t < t_1 \) have no neighbors in \( K \);
- the nodes \( t \) with \( t_i < t < t_{i+1} \) have no neighbors in \( K \) or have exactly as neighbors one of \( C_0, \ldots, C_i \) for \( i \in \{0, \ldots, \ell - 1\} \);
- the nodes \( t \) with \( t_\ell < t \) have no neighbors in \( K \) or have exactly as neighbors one of \( C_1, \ldots, C_\ell \).

We denote its probability as \( p(t_1, \ldots, t_\ell, C_1, \ldots, C_\ell) \), and so we have that

\[
p(t_1, \ldots, t_\ell, C_1, \ldots, C_\ell) = \left[ \prod_{t < t_1} p(t, \emptyset) \right] p(t_1, C_1) \times
\]

\[
\left[ \prod_{t_1 < t < t_2} \left[ p(t, \emptyset) + p(t, C_1) \right] \right] p(t_2, C_2) \times
\]

\[
\left[ \prod_{t_2 < t < t_3} \left[ p(t, \emptyset) + p(t, C_1) + p(t, C_2) \right] \right] p(t_3, C_3) \times \cdots \times
\]

\[
\left[ \prod_{t_{\ell-1} < t < t_\ell} \left[ p(t, \emptyset) + p(t, C_1) + \cdots + p(t, C_{\ell-1}) \right] \right] p(t_\ell, C_\ell) \times
\]

\[
\left[ \prod_{t_\ell < t} \left[ p(t, \emptyset) + p(t, C_1) + \cdots + p(t, C_\ell) \right] \right]
\]

To build a compact expression, let us define \( t_0 \) and \( t_{\ell+1} \) as “virtual nodes” such that \( t_0 < t_1 \) and \( t_\ell < t_{\ell+1} \); and \( C_0 = \emptyset \). Then we have that:

\[
p(t_1, \ldots, t_\ell, C_1, \ldots, C_\ell) = \left[ \prod_{j=1}^{\ell} p(t_j, C_j) \right] \left[ \prod_{j=0}^{\ell} \prod_{t_j < t_{j+1}} \sum_{j=0}^{j} p(t, C_j) \right]. \tag{3}
\]
Note that the terms in the addition \( p(t, \emptyset) + p(t, C_1) + \cdots + p(t, C_j) \) correspond to the probabilities of events that are pairwise disjoint due to the “exactly” conditions in the definition of \( P(x, C) \). Therefore, this addition results in the probability of the union event. It follows that the probability that a 2-path \( E' \) generates the cover \( \{C_1, C_2, \ldots, C_t\} \) and the \( \ell \)-tuple \((t_1, t_2, \ldots, t_\ell)\) with \( t_{E'}(C_i) = t_i \ \forall i \in 1, \ldots, \ell \) is:

\[
p\{(E' : C_{E'} = \{C_1, \ldots, C_t\} : t_{E'}(C_i) = t_i\} = p(t_1, \ldots, t_\ell, C_1, \ldots, C_t).
\]

(4)

Finally, from Eq. (1) and Eq. (4) we have that:

\[
R_K(G, 2) = \sum_{F \subseteq K^{(2)}} \prod_{e \in F} p(e) \prod_{e \in K^{(2)} \setminus F} (1 - p(e)) \sum_{\{C_1, \ldots, C_t\} \subset F} \sum_{\{C_1, \ldots, C_t\} \supseteq K} \sum_{t_1 < t_2 < \cdots < t_\ell < (V \setminus K)^{\ell}} p(t_1, \ldots, t_\ell, C_1, \ldots, C_t). (5)
\]

where the second summation ranges over all covers of \( K \) for \( F \) having a number \( \ell \) of elements between 0 and \( \max(n', 2|K|^{(2)}) \) (recall that the cover is a set and thus its elements are different).

2.5 Computational complexity

The first summation in Eq. (5) has \( 2^{|K|^{(2)}} \) terms. The second summation has no more than \( 2^{2|K|^{(2)}}(|F|! \ell!) \) terms with \( \ell \leq 2^{2|K|^{(2)}} \). The third summation has \( \binom{n'}{2^{2|K|^{(2)}}} \) terms. The product operands involve \( \binom{|K|^{(2)}}{2} \) products. Computing \( p(t_1, \cdots, t_\ell, C_1, \cdots, C_t) \) involves \( n \) products and a number of additions bounded by \((\ell + 1)n\) since, denoting as \( \tau_i \) the position of \( t_i' \) within the ordering of \( V \setminus K \), the number of additions is equal to \( 2(\tau_2 - \tau_1 - 1) + 3(\tau_3 - \tau_2 - 1) + \cdots + \ell(\tau_\ell - \tau_{\ell-1} - 1) + (\ell + 1)(n - \tau_\ell) = -2 - 3 - \cdots - \ell - 2\tau_1 - 2\tau_2 - 3 \cdots - \tau_\ell + (\ell + 1)n < (\ell + 1)n \). Hence the number of elemental operations (additions and products) needed to compute \( R_K(G, 2) \) has order

\[
2^{|K|^2} \left( |K|^2 \right) 2^{2|K|} \binom{n'}{2^{2|K|}} n(2^{2|K|} + 2)
\]

(6)

which is a polynomial in \( n \) of degree \( 2^{2|K|} + 1 \), thus proving that the complexity is polynomial. It is easy to see that an enumeration of the sets \( \mathcal{P}(K^{(2)}) \), \( \{C \supseteq F \} \) and \( (V \setminus K)^{\ell} \) can be done in a number of steps linear in their cardinality; then the computational complexity order of the three summations is the number of terms involved. Now let us see that the complexity is, indeed, linear in \( n \). To do so, it is enough to show that the rightmost summation of Eq. (5) can be computed in a time linear in \( n \), since the remaining summations and product operators in that equation multiply the order by factors that only depend on \( |K| \), thus being constant with regard to \( n \). For simplicity of notation let us assume that \( V \setminus K = \{1, \ldots, n'\} \). Following Eq. (3), the rightmost summation of Eq. (5) coincides with the summation of all products that have the form

\[
p(1, C_{a_1})p(2, C_{a_2}) \cdots p(n', C_{a_{n'}})
\]

where \( a_i \in \{0, 1, \ldots, \ell\} \) (recall that \( C_0 = \emptyset \) and there are \( \ell \) integers \( t_i \) with \( 0 < t_1 < \cdots < t_\ell \leq n' \) such that:
\[ a_t = i \quad \forall i \in \{1, \ldots, \ell\}, \]

- if \( t < t_1 \) then \( a_t = 0 \),
- if \( t_i < t < t_{i+1} \) with \( i \in \{1, \ldots, \ell - 1\} \) then \( a_t \in \{0, \ldots, i\} \),
- if \( t > t_\ell \), then \( a_t \) is any integer between 0 and \( \ell \).

These products can be associated to directed paths in the directed graph defined by:

\[ \vec{G} = (\{1, \ldots, n'\} \times \{0, \ldots, \ell\})^2, \vec{E} \]

\[ \vec{E} = \{( (t, a, b), (t + 1, a', b') ) : t \leq n' - s, t \leq b + 1, 0 \leq a' \leq b + 1, b' = \max(a', b) \} \cup \]

\[ \{( (t, a, b), (t + 1, a', b') ) : n' - \ell \leq t \leq n' - \ell + b, \ell - (n' - t) < a' \leq b + 1, b' = \max(a', b) \} \]

that go from vertex \((1, 0, 0)\) to \((n', \cdot, \ell)\). Each vertex \((t, a, b)\) is associated to the probability \( p(t, C_a) \). The variable \( b \) cumulates the number of nodes of \( t_1, \ldots, t_\ell \) already visited when \( t \) moves from \( t = 1 \) to \( t = n' \); while \( a \) represents the possible sets involved in the events \( p(t, C_a) \). Computing the rightmost summation of Eq. (5) can therefore be done by dynamic programming, proceeding from the vertices with the form \((n', \cdot, \ell)\) downwards to reach \((1, 0, 0)\). In each step, a value \( s(t, a, b) \) is assigned to the vertex \((t, a, b)\) as follows:

\[ s(t, a, b) = \begin{cases} 
  p(t, C_a) & \text{if } t = n', \\
  p(t, C_a) \sum_{(t, a', b') \sim (t + 1, a', b')} s(t + 1, a', b') & \text{if } 0 < t < n'. 
\end{cases} \]

Hence the number of operations for computing the rightmost summation of Eq. (5) will not exceed the number of edges of the graph \( \vec{G} \), which is bounded by \((n' - 1)\ell^2\) times the maximum possible degree \( \ell^2 \), that is \((n' - 1)\ell^4\). This is linear in \( n \), which completes the proof.

### 3 The General Case

The general \( d \)-DCKR is in the class of NP-Hard computational problems, since it is a generalization of the classical \( K \)-terminal reliability, which is NP-Hard. We develop two heuristics inspired in techniques inherited from the classical edge-reliability problem and interpolation theory. The edge-reliability problem is precisely the \( d \)-DCKR problem when \( K = V \) and \( d = |V| - 1 \). The edge-reliability polynomial \( R_V(G, |V| - 1) \) can be written in the following manner:

\[ R_V(G, |V| - 1) = \sum_{i=0}^{m} F_i p^{m-i}(1 - p)^i, \quad (7) \]

where \( F_i \) is the number of connected subgraphs of \( G \) with exactly \( m - i \) edges. Ball and Provan proved that the computation of the \( F \)-vector \( F = (F_0, F_1, \ldots, F_m) \) is in the class of NP-Hard computational problems Provan and Ball (1983). The computation of \( R_V(G, \infty) \) is thus NP-Hard. However, the related literature is vast,
and offers bounds and estimation techniques, as well as exponential algorithms to exactly find that polynomial Satyanarayana and Chang (1983).

In the non-exact approach, the effort is primarily focused on the design of unbiased and time-efficient estimators for \( R_V(G, \infty) \) with reduced variance, and the main mathematical machinery is Monte Carlo-based simulation methods Rubino and Tuffin (2009); Cancela et al. (2012a); Cancela and Khadiri (2003).

A key point is to observe that in the \( d \)-DCKR problem, Expression (7) holds, where now \( F_i^d \) represents the number of \( d \)-connected subgraphs with \( m - i \) edges:

\[
R_K(G, d) = \sum_{i=0}^{m} F_i^d p^{m-i}(1-p)^i,
\]

In Subsection 3.1 we introduce \( F \)-MonteCarlo, which estimates the coefficients \( F_i^d \) directly. A more sophisticated heuristic called \( d \)-Interpol is introduced in Subsection 3.2. The main idea is to find pointwise estimations for the polynomial \( R_K(G, d) \) and finally find the interpolation. Expression (8) is pivotal, and we will use it in both heuristics.

### 3.1 F-MonteCarlo

Recall that \( F_i^d \) represents the number of \( d \)-connected subgraphs with precisely \( m - i \) edges. Consider random graphs with \( m - i \) edges uniformly chosen at random, and denote \( X_1^N, \ldots, X_i^N \) the binary random variable such that \( X_j = 1 \) if and only if the resulting subgraph \( j \in \{1, \ldots, N\} \) is \( d \)-connected (and \( X_j = 0 \) otherwise). The strong law asserts that the average random variable \( \overline{X_i^N} \) converges almost surely to \( E(X_j) \):

\[
E(X_j) = P(X_j = 1) = \frac{F_i^d}{2^{m-i}},
\]

where the last equality uses the fact that all \( 2^{m-i} \) random subgraphs with \( m - i \) are equally likely to occur. Combining equations (8) and (9), it is natural to propose the following random variable:

\[
H(p) = \sum_{i=0}^{m} 2^{m-i} \overline{X_i^N} p^{m-i}(1-p)^i,
\]

Since \( E(2^{m-i} \overline{X_i^N}) = F_i^d \), we get that \( H \) is an unbiased estimator for \( R_K(G, d) \), uniformly in the indeterminate \( p \). Expression (10) suggests the following heuristic:
### Algorithm 1 $R_K(G, d) = F - \text{MonteCarlo}(G, K, d, N)$

1: for $j = 0$ to $m$
2: \hspace{1em} $\text{Sum}_j \leftarrow 0$
3: for $i = 1$ to $N$
4: \hspace{2em} $X^j_i \leftarrow \text{MonteCarlo}(G, K, d)$
5: \hspace{2em} $\text{Sum}_j \leftarrow \text{Sum}_j + X^j_i$
6: end for
7: $X^{\text{Avg}}_N \leftarrow \text{Sum}_j / N$
8: $F^d_j \leftarrow 2^{m-j}X^{\text{Avg}}_N$
9: end for
10: $R_K(G, d) \leftarrow \sum_{i=0}^{m} F^d_i p^i (1 - p)^{m-i}$
11: return $R_K(G, d)$

The reader can notice that the distinguished set of vertices $K$ is considered during $\text{MonteCarlo}(G, K, d)$. In that function, a sample of the random variable $X^j$ is picked, and $X^j_i = 1$ if and only if the resulting subgraph respects the property: “all $K$ sites are $d$-connected”. Additionally, some coefficients can be found beforehand (for instance, $F_0 = 1$ and $F_m = 0$), and we omitted them for the sake of brevity.

#### 3.2 $d$-Interpol

In Roblede et al. (2013), an interpolation-based technique to address the classical edge-reliability problem has been proposed. It exploits the structure of the Hilbert space of $L^2[0,1]$, and combines the efficiency of Newton interpolation with the simplicity of Monte Carlo reliability estimation in selected abscissas in $[0,1]$. The basic idea is to find $m+1$ pointwise estimations for $R_V(G, \infty)$ (with a Monte-Carlo based technique for instance), and to obtain the interpolation polynomial. Here, we extend this concept in order to find the coefficients of the target polynomial $R_K(G, d)$.

### Algorithm 2 $R_K(G, d) = d - \text{Interpol}(G, K, d, N, x)$

1: for $i = 0$ to $m$
2: \hspace{1em} $y_i \leftarrow \text{CrudeMonteCarlo}(x_i, N)$
3: end for
4: y $\leftarrow (y_1, \ldots, y_m)$
5: g $\leftarrow \text{Newton}(x, y)$
6: R $\leftarrow \text{Rounding}(g)$
7: return $R_K(G, d) = R$

As well as $F$-MonteCarlo, $d$-Interpol receives the target graph $G$ with distinguished sites $K$, diameter $d$ and sample size $N$. Additionally, it receives a set of abscissas $x = \{x_0, x_1, \ldots, x_m, x_{m+1}\}$ where $0 \leq x_i \leq x_{i+1} \leq 1$. In the Block of Line 1-3, a pointwise estimation $y_i$ for $R_K(G, d)$ is performed for each abscissa $x_i$, and the information is stored in vector $y$ in Line 4. The polynomial interpolation
takes place in Line 5 (using Newton’s adding point rule). If all estimations were exact, then \( R \) equals \( R_K(G,d) \). However, the introduction of \textit{Crude Monte Carlo} leads to deviations, so the coefficients may not be integers, and a rounding of the coefficients takes effect in Line 6. In the following paragraphs the building blocks of \( d \)-Interpol are detailed.

\textit{Crude Monte Carlo} follows a parallel idea of \( F \)-Monte Carlo, but the probability of operation \( p \) is fixed in a certain value \( x_i \) instead of counting the cardinality of \( F_i \). Now, random graphs are independently picked, where the links operate with probability \( x_i \). As a consequence, an unbiased estimation for \( R_K(G,d) \) is obtained for every abscissa \( x_i \) in the input set \( x \).

Specifically, represent the link states by a binary vector \((X_1,X_2,\ldots,X_m)\) and the network state by the structural indicator random variable \( \Phi : \{0,1\}^m \rightarrow \{0,1\} \), where \( \Phi(X_1,\ldots,X_m) = 1 \) whenever the resulting subgraph with links \( E' = \{e_i : X_i = 1, i = 1,\ldots,m\} \) is \( d \)-connected for the sites \( K \). Suppose we are given a fixed probability of operation \( p \), and we want to find the probability \( \gamma = R_K(G,d)(p) = P(\Phi = 1) \).

\textit{Crude Monte Carlo} (CMC) simulates a sample \( \Phi_1,\ldots,\Phi_N \) of \( N \) independent identically distributed variables respecting the structural law \( \Phi \), and proposes the estimator \( \overline{\Phi}_n \) for \( P(\Phi = 1) \):

\[
\overline{\Phi}_N = \frac{1}{N} \sum_{i=1}^{N} \Phi_i. \tag{11}
\]

As corollary from Kolmogorov’s Strong Law, the estimator \( \overline{\Phi}_n \) is consistent for \( \gamma \). Moreover, \( E(\overline{\Phi}_N) = \gamma \) for all \( N \geq 1 \), so it is also unbiased. Assume \( N \) is sufficiently large to apply the Central Limit Theorem. Therefore, a confidence interval at level \( \alpha \) for \( R_G(p) \) is centered at \( \overline{\Phi}_n \) and its radius is approximately equal to:

\[
\delta = \sqrt{\overline{\Phi}_N(1 - \overline{\Phi}_N) t_{\alpha/2}(N - 1)} \sqrt{\frac{1}{N}} \tag{12}
\]

where \( P(T > t_{\alpha/2}(n-1)) = \frac{\alpha}{2} \) for a t-Student random variable \( T \) with \( n - 1 \) degrees. Observe that if we could choose the probability \( p \) such that \( R_K(G,d)(p) \approx \frac{1}{2} \), we avoid corner effects coming from a rare event simulation Rubino and Tuffin (2009). In case \( \overline{\Phi}_N \approx \frac{1}{2} \), the radius is roughly:

\[
\delta \approx \frac{z_{\alpha/2}}{2\sqrt{N}}, \tag{13}
\]

where \( P(>z_{\alpha/2}) = \frac{\alpha}{2} \) for a normal standard variable \( Z \). Equation (13) states that the global error is inversely proportional to the square root of the sample size \( N \).

Now, suppose we choose \( m + 1 \) probabilities of operation \( x = \{x_1,\ldots,x_m\} \) such that \( R_K(G,d)(x_1) \approx \frac{1}{2} \) (or at least, \( \min\{1 - R_K(G,d)(x_1), R_K(G,d)(p_1)\} > \delta \) for some \( \delta > 0 \) large enough. A primitive bipartition in the set \([0,1]\) makes this assumption feasible. The idea of \( d \)-Interpol is to obtain the target polynomial \( R_K(G,d) \) by means of the resulting CMC estimations \( y_i = \overline{\Phi}_N(p_i) \), with classical polynomial interpolation.
There exist several ways to find the unique interpolant. Vandermonde’s method is conceptually simple, but an ill-conditioned linear system must be faced. Lagrange polynomials are easy to write, but hard to handle. We will see that Newton interpolation method is simultaneously simple and easy to evaluate. Last but not least, an error analysis is also feasible using Newton interpolation. The reader can find a thorough exposition of classical interpolation theory in the book Phillips (2003). By the previous reasons, we will focus on the Newton interpolation method. This method iteratively constructs the final interpolation including in each iteration a new point \((x_i, g(x_i))\). In fact, if we are given a set of points \((x_i, y_i)\) and the \(i\)-th polynomial \(g_i\) interpolates the first \(i\) points, then the new polynomial \(g_{i+1}(x) = g_i(x) + c_{i+1} \prod_{1 \leq j \leq i} (x - x_j)\) respects \(g_{i+1}(x_k) = g_i(x_k)\) for all \(k \leq i\), and \(c_{i+1}\) is found choosing \(g_{i+1}(x_{i+1}) = y_{i+1}\). The base step is the constant \(g_1(x) = y_1\).

If \(g = g_m\) is the polynomial interpolation of a certain function \(h\) in the points \(\{(x_i, y_i)\}_{i=1,...,m}\) by means of Newton method (where \(y_i = h(x_i)\)), the global error can be found exactly for a given \(x \in [x_1, x_m]\) Isaacson and Keller (1994):

\[
h(x) - g(x) = [x_1, x_2, \ldots, x_m, x](h) \prod_{i=1}^{m} (x - x_i),
\]

where the operator \([x_1, x_2, \ldots, x_n](f)\) is recursively defined by \([x_1]f = f(x_1)\) and

\[
[x_1, \ldots, x_k]f \equiv \frac{[x_2, \ldots, x_k](f) - [x_1, \ldots, x_{k-1}](f)}{x_k - x_1}.
\]

Therefore, we can find the order of a uniform bound for the worst error \(\{|h(x) - g(x)|\}_{x \in [0,1]}\) in terms of \(N\) if we further neglect the rounding errors. Indeed, by (15), the propagation error can be duplicated in the worst case in each step, so the new radius is of order \(\delta' = 2^{m+1} \delta\). As a consequence, we get that \(\delta' = O(2^{m}/\sqrt{N})\), which suggests the sampling size \(N\) should increase exponentially with respect to the size of the graph. This suggests the hardness of the problem we are addressing. However, we will contrast this interpolation-based heuristic with \(F\)-MonteCarlo for graphs with limited size.

4 Numerical Results

The effectiveness of both heuristics is measured on the lights of two generalized Petersen graphs: Petersen \(G(5,2)\) and Dodecahedron \(G(10,2)\), shown in Figures 3 and 2 respectively.

The diameter of \(G(5,2)\) is \(D(G(5,2)) = 2\), and the reader can check that \(G(5,2)\) is both 2-critical and 3-critical (i.e. its diameter is increased to \(D = 4\) under an arbitrary link deletion). In particular, if an arbitrary link \(e = (xy)\) is deleted, the incident sites \(x\) and \(y\) do not have a path with less than 4 hops. On the other hand, \(D(G(10,2)) = 5\), and it is not 5-critical.

By counting, the coefficients \(F_i^D\) can be obtained in both graphs. In Petersen’s graph:

\[
R_{G(5,2)}(V, 4) = p^{15} + 15p^{14}(1 - p) + 105p^{13}(1 - p)^2 + 385p^{12}(1 - p)^3 +
\]

\[
= 480p^{11}(1 - p)^4 + 72p^{10}(1 - p)^5 + 10p^9(1 - p)^6,
\]
In order to have a faithful analysis of performance for $d$-Interpol, both polynomials $R_{G(5,2)}(V, 4)$ and $R_{G(10,2)}(V, 5)$ must be re-written in their general form: $\sum_{i=0}^{m} P_d^i p^i$, where the coefficients $P_d^i$ are linearly related with the set $F^D_i$:

$$P_{m-k}^d = \sum_{i=k}^{m} F^d_i \binom{i}{k} (-1)^{i-k}.$$  

Table 2 presents the exact coefficients $P_d^i$ for Petersen graph and the corresponding estimation obtained by $d$-Interpol (null coefficients are not shown).
The reader can appreciate the enormous gaps between the exact coefficients and the ones provided by $d$-Interpol, even with large sample sizes. We suspect this numerical unstability is related with corner effects. Indeed, when $p$ is lower but close to 1, it is rare to pick a random graph that is not $d$-connected (analogously, when $p$ is close to 0 it is rare to pick $d$-connected graphs). This interpolation-based heuristic is unstable in Dodecaedron as well, and its adjustment deserves further research (see Section 5 for a discussion of possible enhancements).

We will now study the performance of $F$-MonteCarlo on the lights of both graphs. For the sake of simplicity, we will directly compare the coefficients $F_i^d$ in this case (null coefficients are not shown).

$F$-MonteCarlo is able to match the coefficients for Petersen graph with success, even when the sample size is reduced. Moreover, When $N = 10^6$ all coefficients are correct but two, with small gaps.

Again, $F$-MonteCarlo is able to guess the lowest coefficients $F^5_i$, $i \in \{1,2,3\}$ for Dodecaedron graph as well, and presents small relative errors in the remaining coefficients.

## Discussion

The 2-DCKR problem is linearly solvable, so the computation of $R_K(G,2)$ is polytime for $|K|$ fixed (proof in Section 2). In general, the computational
complexity of evaluating $R_K(G, d)$ is NP-hard, since this measure subsumes the classical $K$-terminal reliability $R_K(G, n-1)$, known to belong to this complexity class Provan and Ball (1983). An elementary calculation leads to an exact expression for $R_K(G, d)$ when $|K| = d = 2$ (the two-terminal case with no more than two hops). However, when $d \geq 3$ the evaluation of $R_K(G, d)$ is in the class of NP-Hard problems Cancela and Petingi (2004). The computational complexity of the evaluation of $R_K(G, d)$ for a variable $K$ is still an open problem, even when $d = 2$.

We address the general $d$-DCKR counting subgraphs with a naive heuristic called $F$-MonteCarlo, and looking for the target polynomial $R_G(K, d)$ with Newton interpolation approach, called $d$-Interpol. Even though the former is simple, it performs better than the latter. We suspect the performance of $F$-MonteCarlo will deteriorate with the graph size, unless the sample size $N$ is increased correspondingly. The reason is that the number of subgraphs is exponentially increasing with $m = |E|$. On the other hand, $d$-Interpol deserves further research. Indeed, the selection of the abscissas should simultaneously minimize the interpolation error and avoid corner effects (to know, the main problem of rare event simulation Rubino and Tuffin (2009)). Moreover, MonteCarlo has been used as the pointwise estimation machinery, but in the classical edge reliability problem several alternatives have been proposed, and an adaption should be considered Cancela et al. (2013, 2012b); Hui et al. (2005).

6 Conclusions and Ongoing Work

In this article we have proved that computing the 2-DCKR is polytime with respect to the number of nodes of the network, provided that $|K|$ is any fixed parameter. Recall that it was already known that for $d > 2$ the problem is NP-hard. The case where $|K|$ is also considered as an input for the complexity analysis seems to be also an NP-hard problem; although this has not yet been demonstrated.

It is worth to observe that the problem $P_1$ of computing the 2-DCKR with $K = V$ and all edge reliabilities equal to $p = 1/2$ is equivalent to the problem $P_2$ of counting all the partial graphs $(V, E' \subseteq E)$ of $G$ with diameter two. This is due to the fact that under such hypothesis every partial graph has the same probability of occurrence ($2^{-n}$) and each partial graph is $2-V$-connected if and only if it has diameter two; it follows that $R_V(G, 2)$ equals the number of such partial graphs divided by $2^n$. Then, proving that the problem $P_2$ is NP-hard would suffice to prove that the 2-DCKR with $K = V$ is NP-hard too; since $P_2$ is a special case of the latter (where all edges are assigned reliabilities equal to 1/2). Furthermore, it would follow that the 2-DCKR with $K$ as a free input for the complexity would be NP-hard, since so it is its particular case 2-DCKR with $K = V$.

In this opportunity, we show a rough counting of $d-K$-connected subgraphs by means of MonteCarlo techniques, and $F$-MonteCarlo has an outstanding performance, at least for graphs of limited size. On the other hand, $d$-Interpol offers an attractive connection between classical interpolation theory and network reliability, but deserves adjustments and further research.
Currently, our efforts head towards proving that $P_1$ is an NP-hard problem, and studying closer the connection between polynomial interpolation and reliability of coherent systems.

References


