Final version appears in Networks, 22 (1992), pp. 369-384.

Algorithms for K-Terminal Reliability Problems with Node Failures

Ehab S. Elmallah
Department of Computing Science
University of Alberta,
Edmonton, Canada

Abstract

Consider a distributed processing system with a set K of sites that can either cooperate in computing a function or hold resources required by other sites. The system is implemented using a communication network with unreliable nodes. Two simplified reliability problems then arise. In the first problem, we are interested in computing the probability that every operational pair of sites in K can communicate with each other. This problem is known to be #P-complete. In the second problem, the sites in K are service centers. Our reliability measure is the probability that every operational site in the network is connected to at least one operational service center. In this paper, we define the class of t-polygon graphs, $t \geq 3$, as the intersection graphs of straight line chords in a convex t-gon. Hence, any t-polygon graph is a circle graph. We show that both problems admit polynomial time solutions when the underlying graph of the network is restricted to a t-polygon graph, for a fixed t.

1 Introduction

In this paper we deal with two network connectedness reliability problems that arise in the reliable assignment of resources to nodes in a distributed processing system. Our model of the network is that of a probabilistic graph $G = (V, E)$ where each vertex x fails independently of other vertices with probability p_x.

In the first problem, every operational pair in a set K, $K \subseteq V(G)$, of terminals requires to communicate with each other. The reliability of the network is the probability that there is a subgraph G' of G whose nodes are operational and all operational nodes in K lie in a single component of G'. We adopt the convention that if all nodes in K fail then any subgraph of G is operational. We call this problem the K-*terminal* reliability problem with unreliable nodes (denoted $UN\text{–}Rel_K$ for short). In the second problem, a collection of $|K|$ identical resources is assigned to a set of $|K|$ nodes, called *service centers* hereafter. Here, a subnetwork G' is functional if every surviving node can gain access to an operational service center. Hence, in our simplified problem, we seek the probability that each operational node lies in one component with an operational service center. Call this latter problem the K-*resource* reliability problem. In either problem, the reliability of an n-vertex graph G whose vertices have the same probability p of operation is given by the polynomial $\sum_{i=0}^{i=n} F_i p^i (1-p)^{n-i}$ where F_i is the number of i-vertex operational subgraphs of G.

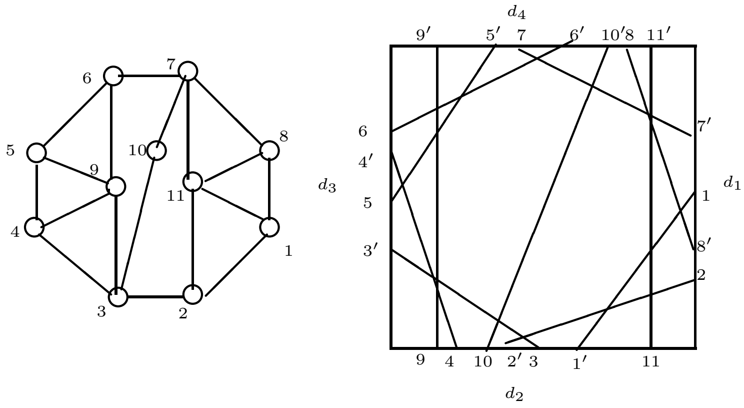
An overview of some related problems and results follows now in order. The K-terminal reliability problem where edges fail independently and nodes are perfectly reliable (denoted $UE\text{–}Rel_K$) has been studied extensively in the literature for directed and undirected graphs. Valiant [11] has shown that the $UE\text{–}Rel_K$ problem, $|K| = 2$, is #P-complete. Ball [2] and Provan and Ball [9] have proven similar complexity results assuming different forms of reliability evaluations and approximations. Subsequently, Provan [8] has shown that the problem remains NP-hard even if G is a planar graph or an acyclic graph and $|K| = 2$.

Similar complexity results can be shown for the $UN\text{–}Rel_K$ problem using a transformation that replaces each unreliable edge with two reliable edges incident to a new unreliable node. Recently, AboElFotoh and Colbourn [1] have shown that a variant of the problem where nodes in K are perfectly reliable remains NP-hard for chordal graphs and comparability graphs. Later, Sutner, Satyanarayana and Suffel [10] have proven that the $UN\text{–}Rel_{V(G)}$ problem (called the residual node connectedness reliability problem in [10]) is NP-hard for split graphs and for bipartite planar graphs. The complexity of the K-resource problem, however, seems to be open.

In view of the apparent intractability of virtually all of the above K-terminal reliability problems, many researchers focused on developing efficient algorithms for restricted classes of graphs and efficiently computable lower and upper bounds (see for example [4] and the references therein). One particularly important class of reliability bounding techniques for the $UE\text{–}Rel_K$ problem is due to [3]. Unfortunately, such techniques do not apply to the $UN\text{–}Rel_K$ (or the K-resource) problem due to a fundamental difference in the combinatorial structure of the *pathsets* (operational subgraphs) of the two problems. Namely, the pathsets in the $UE\text{–}Rel_K$ problem form a *hereditary* family. That is, if $G' \subseteq G$ is operational then any supernetwork G'', $G' \subseteq G'' \subseteq G$, is also operational. This property does not hold for our problems.

In this paper, we define t-polygon graphs, $t \geq 3$, to be the intersection graphs of straight line chords in a convex t-gon. Hence, permutation graphs form a proper subset of 3-polygon graphs. Moreover, circle graphs $= \bigcup_{t=3}^{\infty} t$-polygon graphs. We show that the $UN\text{–}Rel_K$ and the K-resource reliability problems on t-polygon graphs can be solved in $O(n^{f(t^2)})$ and $O(n^{g(t^2+|K|)})$ time, respectively, where f and g are simple polynomials. Our first result generalizes a recent result of [5] that solves the residual connectedness reliability ($UN\text{–}Rel_{V(G)}$) problem in $O(n^3)$ time on permutation graphs. In general, the results are useful in obtaining bounds on the reliability of a network that can be approximated (either by adding edges or deleting edges) by a k-gon graph with a small k.

The remaining part of this paper is organized as follows. In section 2, we fix some definitions and notations. Section 3 outlines the general strategy for computing the reliability measures; the specific ingredients of the algorithms are given in sections 4 and 5. Finally, we draw some conclusions in section 6. Throughout our presentation, we use the graph in Figure 1 as a runing example.



$C_1 = \{(1,1'),(2,2')\}$

$C_2 = \{(3,3'),(4,4')\}$

$C_3 = \{(5,5'),(6,6')\}$

$C_4 = \{(7,7'),(8,8')\}$

$C_5 = \{(9,9'),(10,10'),(11,11')\}$

$C_6 = \phi$

$ext_l(C_5) = \{(9,9')\}$

$Next_l(C_5) = \{(3,3'),(6,6')\}$

$ext_r(C_5) = \{(11,11')\}$

$Next_r(C_5) = \{(2,2'),(7,7')\}$

(a) (b) (c)

Figure 1.

2 Definitions and Notations

The intersection graph $G = (V, E)$ of a set of chords C inside a t-gon P_t has a vertex v_i for each chord (i, i') and an edge (v_i, v_j) if and only if (i, i') intersects (j, j'). The sides of P_t are labelled $d_1, d_2, \ldots, d_t$ as they appear in a clockwise traversal of P_t. The diagram of C inside P_t is called a *polygon* diagram of G. For any two points i and i' on a side d of the t-gon P_t we denote the interval on d between (and including) i and i' by $[i, i']$. We also denote the particular side of P_t that contains a given point i by $d(i)$.

We view the set C as a union of $\binom{t}{2}$ disjoint sets, $C_1, C_2, \ldots, C_{\binom{t}{2}}$, called *chordal classes*. Each chordal class C_i is defined by a unique pair of sides that contains the endpoints of all chords in that particular class. Let $I = \{1, 2, \ldots, \binom{t}{2}\}$ be the set of indices of such classes. The following definitions identify certain structures related to chordal classes (see also Figure 1).

Definition 1 (*extreme points and chords*). Let C_i, $i \in I$, be a chordal class defined by the sides d_x and d_y of P_t and let W_i be a nonempty subset of C_i. The *extreme* points of W_i is a collection i_1, i_2, i_3 and i_4 of end-points of chords in W_i such that $i_1, i_2 \in d_x$ and $i_3, i_4 \in d_y$ (it is possible that $i_1 = i_2$ and/or $i_3 = i_4$) and if $(l, l') \in W_i$ then $l \in [i_1, i_2]$ and $l' \in [i_3, i_4]$. We also call the set of chords incident with the extreme points the *extreme* chords and denote it by $ext(W_i)$. ■

Definition 2 (*windows and casts*). Let $W_i \subseteq C_i$ be as above. Call W_i a *window* in C_i if W_i contains all chords in C_i whose two endpoints lie on d_x and d_y between the extreme points of W_i. In addition, we regard the empty set as a window in C_i. A *cast* of G is a sequence $\mathcal{W} = (W_1, W_2, \ldots, W_{|I|})$ of windows where $W_i \subseteq C_i$, for $i \in I$. Let $ext(\mathcal{W}) = \bigcup_{W_i \in \mathcal{W}} ext(W_i)$. ■

In the following definition, we partition the set of extreme points of a nontrivial window W_i into an extreme-left set and an extreme-right set. Roughly speaking, this is done be choosing two arbitrary reference points (denoted m_x and m_y) that lie in the middle of the extreme points of W_i and then splitting the

k-gon into a left half and a right half.

Definition 3 (*the extreme-left and extreme-right chords*). Let $W_i \subseteq C_i$, $|W_i| \geq 2$, be a window where C_i is defined by the two polygon sides d_x and d_y, $x > y$. Let $i_1, i_2 \in d_x$ and $i_3, i_4 \in d_y$ be the extreme points of W_i. Let $m_x \in [i_1, i_2]$ and $m_y \in [i_3, i_4]$ be two points that are not end-points of any chord in the diagram. Viewing P_t as a cycle on t vertices, let $\mathrm{P}_{t(x \to y)}$ and $\mathrm{P}_{t(y \to x)}$ be the two paths encountered in a clockwise traversal of P_t from m_x to m_y and from m_y to m_x, respectively. Call the extreme points of W_i lying on $\mathrm{P}_{t(x \to y)}$ (similarly, $\mathrm{P}_{t(y \to x)}$) the *extreme-right* (respectively, *extreme-left*) points of W_i.

An *extreme-left (extreme-right) chord* is then an extreme chord with one left-extreme-point (respectively, right-extreme-point) and the other end-point is either unlabelled or a left-extreme-point (respectively, right-extreme-point). Denote the extreme-left and extreme-right chords of W_i by $ext_l(W_i)$ and $ext_r(W_i)$, respectively.

In all other cases, (where $|W_i| = 1$ *or* $|W_i| \geq 2$ and some extreme chord of W_i has its two end-points labelled extreme-left and extreme-right) the subgraph induced by $ext(W_i)$ is connected. Here, the distinction between left and right is immaterial; we set $ext(W_i) = ext_l(W_i) = ext_r(W_i)$ to be the left/right chords. ■

Where the two halves $\mathrm{P}_{t(x \to y)}$ and $\mathrm{P}_{t(y \to x)}$ are defined by the context and i is a given end-point of some chord, we write $h(i)$ and $oh(i)$ for the half of P_t containing i and the opposite half (that does not contain i), respectively. Given a cast $\mathcal{W}$, we next associate with each nonempty window $W_i \in \mathcal{W}$ a set of chords $Next(W_i)$ having the following properties: (i) $Next(W_i) \subseteq ext(\mathcal{W}) \setminus ext(W_i)$ and (ii) if α is a chord in W_i that intersects some chord in some other window $W_j \in \mathcal{W}$, $i \neq j$, then α intersects some chord in $Next(W_i)$. Hence, the set $Next(W_i)$ summarizes the adjacency relation between W_i and the remaining chords in $\mathcal{W}$.

Definition 4 (*the extreme neighbours*). Let W_i be as in the previous definition. Denote by $dom(\mathcal{W} \backslash W_i)$ the subset of $ext(\mathcal{W}) \setminus ext(W_i)$ where each chord dominates (i.e. intersects) some chord in W_i. In addition, let $W'_i \subseteq W_i$ be the set dominated by $dom(\mathcal{W} \backslash W_i)$. We now identify a set $Next(W_i)$ that contains a (nearly minimal) subset of $dom(\mathcal{W} \backslash W_i)$ that dominates W'_i. Clearly, if $dom(\mathcal{W} \backslash W_i) = \phi$ then $Next(W_i) = \phi$. In addition, if $dom(\mathcal{W} \backslash W_i)$ has a chord (l, l') that dominates W_i then set $Next(W_i) = (l, l')$. Otherwise, every chord in $dom(\mathcal{W} \backslash W_i)$ has exactly one end-point on either $[i_1, i_2]$ or $[i_3, i_4]$.

Let the two *halves* ($\mathrm{P}_{t(x \to y)}$ and $\mathrm{P}_{t(y \to x)}$) of P_t be as in Definition 3. For each possible point $i_k \in (i_1, i_2, i_3, i_4)$ let $dom(\mathcal{W} \backslash W_i)_{d(i_k), oh(i_k)}$ be the set of chords in $dom(\mathcal{W} \backslash W_i)$ with one end-point on the side $d(i_k)$ and the other on the half $oh(i_k)$. Note that $dom(\mathcal{W} \backslash W_i)_{d(i_k), oh(i_k)}$ does not depend on any exact choice of m_x and m_y in Definition 3. Now, for $k = 1, \ldots, 4$, if $dom(\mathcal{W} \backslash W_i)_{d(i_k), oh(i_k)} \neq \phi$, let i'_k be the closest end-point to i_k of a chord $(i'_k, i''_k) \in dom(\mathcal{W} \backslash W_i)_{d(i_k), oh(i_k)}$. Assign the possible chords incident to i'_k, $k = 1, \ldots, 4$, to $Next(W_i)$. In addition, let $Next_l(W_i)$ (also, $Next_r(W_i)$) be the subset of $Next(W_i)$ where each chord intersects some chord in $ext_l(W_i)$ (respectively, $ext_r(W_i)$). See Figure 1 for an example. ■

We also need the following definitions.

Definition 5 (*the representative graph of a cast*). The representative graph of a cast $\mathcal{W}$, denoted $R_{\mathcal{W}}$, has two vertices corresponding to $ext_l(W_i)$ and $ext_r(W_i)$ for each $W_i \in \mathcal{W}$ where $ext_l(W_i) \neq ext_r(W_i)$. Otherwise, (if $ext_l(W_i) = ext_r(W_i)$) then $R_\mathcal{W}$ has one vertex corresponding to $ext(W_i)$. Hence, $R_\mathcal{W}$ has at most $2|\mathcal{W}|$ vertices; some vertices may represent empty sets. In addition, $(x, y) \in E(R_\mathcal{W})$ if and only if some chord in the set represented by x intersects a chord in the set represented by y. ■

Definition 6 (*p-subgraphs*). Given a probabilistic graph G, we define a *p-subgraph* of G to be any induced subgraph $G' \subseteq G$ where $V(G) \setminus V(G')$ contains only unreliable vertices. We interpret that all vertices in G' are operational and the remaining are failed. ■

We recall that if G is a permutation graph whose vertices are labelled $(1, 2, \ldots, n)$ then G is described by some permutation π of $\{1, 2, ..., n\}$. We write π as a sequence $(\pi_1, \pi_2, ..., \pi_n)$ and let π_i^{-1} be the position in the sequence where the number i can be found. By definition, $(i, j) \in E(G)$ if and only if $i > j$ and $\pi_i^{-1} < \pi_j^{-1}$.

Throughout our presentation of the algorithms we find it convenient to initialize some variables to a special value denoted $\perp$ (called *bottom*). A variable assigned this value may appear in any of the four basic arithmetic operations, the two boolean operations $\vee$ and $\wedge$ or the two functions $min()$ and $max()$. In each case, the value $\perp$ is treated as the identity of the particular operation used. For instance, if $x = \perp$ and $y \geq 1$ then $x = \infty$ when evaluating $min(x, y)$. Testing whether or not a variable has the value $\perp$ is done in a straightforward way. However, the logical result of all comparisons using $<$ or $>$ and a variable assigned the special value $\perp$ is *false*. Using the above convention results in a more concise description of the algorithms.

3 Outline of the Algorithms

We now outline the general structure of the main algorithms. Given an instance of the $UN\text{–}Rel_K$ problem or the K-resource reliability problem defined on a graph G that is represented by a polygon diagram. Let $\mathcal{W}$ be a cast in the diagram. Denote by $G_\mathcal{W}$ the subgraph of G induced by $\mathcal{W}$. Assuming that all vertices in $ext(\mathcal{W})$ are perfectly reliable, let $Rel(G_\mathcal{W})$ be the probability of obtaining an operational subgraph of $G_\mathcal{W}$. Summing over all possible distinct casts in a polygon diagram of G we get

$$Rel(G) = \sum_{\mathcal{W}} \left(Rel(G_\mathcal{W}) \prod_{i \in ext(\mathcal{W})} p_i \prod_{j \notin G_\mathcal{W}} (1 - p_j) \right). \tag{1}$$

For the purpose of computing $Rel(G_\mathcal{W})$, we associate with each window $W_i \in \mathcal{W}$ the graph $G_{\mathcal{W},i}$ induced by $W_i \cup Next(W_i)$ in G.

The strategy is to compute $Rel(G_\mathcal{W})$ from certain probabilities associated with the subgraphs in $\mathcal{G}_\mathcal{W} = \{G_{\mathcal{W},1}, G_{\mathcal{W},2}, \ldots, G_{\mathcal{W},|I|}\}$. We first specify (in the appropriate sections) a set of necessary conditions for any p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$, $i \in I$, to be in some operational subnetwork of G. For the convenience of presenting the strategy, we split the computations into two phases. In the first phase, we define a set $\mathcal{S}^i$ of *short* state-vectors. Each p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ satisfying the specified necessary conditions is characterized by a unique state-vector in $\mathcal{S}^i$. And many subgraphs of $G_{\mathcal{W},i}$ may have the same state-vector.

Given a function $f^i : \mathcal{G}_\mathcal{W} \rightarrow \mathcal{S}^i$ we denote by $\mathcal{F}(G_{\mathcal{W},i}, f^i)$ the set of p-subgraphs of $G_{\mathcal{W},i}$ where each member is in state $f^i(G_{\mathcal{W},i})$. Likewise, let $Pr(G_{\mathcal{W},i}, f^i)$ be the probability of obtaining a subgraph in

$\mathcal{F}(G_{\mathcal{W},i}, f^i)$. In addition, let $\mathcal{F}(G_{\mathcal{W}}, f^i)$ be the set $\{H|\ H \subseteq G_{\mathcal{W}}\ \ and\ H \cap G_{\mathcal{W},i} \in \mathcal{F}(G_{\mathcal{W},i}, f^i),\ i \in I\}$. The probability $Pr(G_{\mathcal{W}}, f^i)$ of obtaining a p-subgraph in such a family is just $\prod_{i\in I} Pr(G_{\mathcal{W},i}, f^i)$.

For each reliability problem in section 1, we define $\mathcal{S}^i$ such that either all subgraphs in $\mathcal{F}(G_{\mathcal{W}}, f^i)$ are operational or failed. To determine their status, we define (in the appropriate section) a function $\Theta(G_{\mathcal{W}}, f^i)$ that evaluates to 1 if and only if all such subgraphs are operational. It then follows that

$$Rel(G_{\mathcal{W}}) = \prod_{\substack{f^i\ s.t. \\ \Theta(G_{\mathcal{W}}, f^i) = 1}} Pr(G_{\mathcal{W}}, f^i). \tag{2}$$

To compute the factors $Pr(G_{\mathcal{W}}, f^i)$ in 2, for all possible functions f^i, we define in the second phase a set $\mathcal{S}^{ii}$ of *long* state-vectors. If $f^{ii} : \mathcal{G}_{\mathcal{W}} \rightarrow \mathcal{S}^{ii}$ then we use the notations $\mathcal{F}(G_{\mathcal{W},i}, f^{ii})$, $Pr(G_{\mathcal{W},i}, f^{ii})$, $\mathcal{F}(G_{\mathcal{W}}, f^{ii})$ and $Pr(G_{\mathcal{W}}, f^{ii})$ as above. Here, each p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ is characterized by a unique vector in $\mathcal{S}^{ii}$ (regardless of whether or not $G'_{\mathcal{W},i}$ is a candidate to appear in some operational subnetwork of G). Thus, for any two states β_1 and β_2 in $\mathcal{S}^{ii}$ the two sets $\mathcal{F}(G_{\mathcal{W},i}, \beta_1)$ and $\mathcal{F}(G_{\mathcal{W},i}, \beta_2)$ are disjoint.

Subsequently, we show how to compute $Pr(G_{\mathcal{W},i}, \alpha)$ for any state $\alpha \in \mathcal{S}^i$ given the set $\{Pr(G_{\mathcal{W},i}, \beta)|\beta \in \mathcal{S}^{ii}\}$. To this effect, we associate with each short state-vector α, $\alpha \in \mathcal{S}^i$, a set $(\mathcal{S}^{ii})_\alpha$ (the restriction of $\mathcal{S}^{ii}$ to α) where $(\mathcal{S}^{ii})_\alpha =\ \{\beta|\ \beta \in \mathcal{S}^{ii}\ \ and\ \mathcal{F}(G_{\mathcal{W},i}, \beta) \subseteq \mathcal{F}(G_{\mathcal{W},i}, \alpha)\}$. Consequently, we have

$$Pr(G_{\mathcal{W},i}, \alpha) = \sum_{\beta \in (\mathcal{S}^{ii})_\alpha} Pr(G_{\mathcal{W},i}, \beta). \tag{3}$$

Finally, we deal with the following problem. Given an n-vertex graph $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$ compute $\{Pr(G_{\mathcal{W},i}, \beta)|\ \beta \in \mathcal{S}^{ii}\}$. Note that, if $|Next(W_i)| \geq 2$ then $G_{\mathcal{W},i}$ may not be a permutation graph. Nevertheless, we exploit the intersection structure of $G_{\mathcal{W},i}$ in the following way. We label the vertices in $G_{\mathcal{W},i}$ by $1, 2, \ldots, n$ and define a permutation $\pi(G_{\mathcal{W},i})$ of $\{1, 2, \ldots, n\}$ such that the permutation graph $H_{\mathcal{W},i}$ corresponding to $\pi(G_{\mathcal{W},i})$ satisfies: $Pr(H_{\mathcal{W},i}, \beta) = Pr(G_{\mathcal{W},i}, \beta)$ for every $\beta \in \mathcal{S}^{ii}$. We then devise an algorithm to solve the problem on permutation graphs. This completes the outline of the algorithms.

4 The K-terminal Reliability Problem

Following the above discussion, we start by defining a set $\mathcal{S}^i$ of short state-vectors for the $UN\text{–}Rel_K$ problem. Let $\mathcal{W}$ be a cast and $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$ as above. Denote by $K_{\mathcal{W},i}$ the terminals in $G_{\mathcal{W},i}$. A general p-subgraph $G'_{\mathcal{W},i}$ of $G_{\mathcal{W},i}$ may have one or more components, say $N_1, N_2, \ldots, N_\ell$. If $ext_l(W_i) \neq ext_r(W_i)$ then we adopt the convention $ext_l(W_i) \subseteq V(N_1)$ and $ext_r(W_i) \subseteq V(N_\ell)$. Otherwise, $G'_{\mathcal{W},i}$ has only one component N_1. In fact, since $ext(W_i) \bigcup Next(W_i)$ are assumed to be perfectly reliable in $G_{\mathcal{W},i}$ then $ext(W_i) \bigcup Next(W_i) \subseteq G'_{\mathcal{W},i}$. Furthermore, it is easy to check that, in $G_{\mathcal{W},i}$, the subgraph induced by $ext_l(W_i) \bigcup Next_l(W_i)$ (similarly, $ext_r(W_i) \bigcup Next_r(W_i)$) is connected. Hence, the above convention implies $Next_l(W_i) \subseteq N_1$ and $Next_r(W_i) \subseteq N_\ell$. We then have

Lemma 4.1 *Let $G'_{\mathcal{W},i}$ be a p-subgraph of $G_{\mathcal{W},i}$. In addition, let $K'_{\mathcal{W},i}$ be the operational terminals in $G'_{\mathcal{W},i}$. Then $G'_{\mathcal{W},i}$ is a candidate to appear in some operational subnetwork of G if (i) $K'_{\mathcal{W},i} = \phi$ or (ii) $K'_{\mathcal{W},i} \neq \phi$ and $K'_{\mathcal{W},i}$ appears in exactly one component of $G'_{\mathcal{W},i}$. In addition, if $1 \leq |K'_{\mathcal{W},i}| < |K|$ and $N_1 \neq N_\ell$ then either $K'_{\mathcal{W},i} \subseteq N_1$ or $K'_{\mathcal{W},i} \subseteq N_\ell$.*

Proof. It suffices to show *that* if N_a and N_b, $N_a \neq N_b$, are two different components in $G'_{\mathcal{W},i}$ then they remain isolated from each other in any graph G', $G'_{\mathcal{W},i} \subseteq G' \subseteq G$. Hence, if a and b are two operational terminals where $a \in N_a$ and $b \in N_b$ then a and b remain disconnected from each other in any network $G' \supseteq G'_{\mathcal{W},i}$ and consequently G' is not operational. To see *that*, assume to the contrary that $G' \subseteq G$ and a is connected to b by a path P in G'. Write P as $(a, \ldots, a', \ldots, b', \ldots, b)$ where a' (b') is the closest vertex to a (respectively, b) on P and a', $b' \notin W_i$ (hence, $a = a'$ if $a \in Next(W_i)$).

Now, the two chords representing a' and b' can be written as (a'_{in}, a'_{out}) and (b'_{in}, b'_{out}), respectively, where a'_{in} and b'_{in} lie between the extreme points of W_i on the two polygon sides d_x and d_y defining W_i. By the geometry of the polygon diagram, if a'_{out} and b'_{out} lie on two different halves of P_t (cf. Definition 3) then P can not possibly exist. Hence, a'_{out} and b'_{out} lie on the same half of P_t, say the *left* half. By definition, every vertex in W_i that is dominated by a' or b' is also dominated by some vertex in $Next_l(W_i)$. By the properties of the polygon diagram, the subgraph induced by $Next_l(W_i) \bigcup ext_l(W_i)$ is connected. Hence, $N_a = N_b$, a contradiction. ■

Define $\mathcal{S}^i$ to be the set $\{\phi, M, L, R, LR\}$. The state α of $G'_{\mathcal{W},i}$ is defined as follows:

1. $\alpha = \phi$ if $K'_{\mathcal{W},i} = \phi$. In the remaining cases $K'_{\mathcal{W},i} \neq \phi$.
2. $\alpha = M$ if $K'_{\mathcal{W},i} \subseteq N_i \subseteq G'_{\mathcal{W},i}$ where $N_i \neq N_1 \neq N_\ell$.
3. $\alpha = L$ if $K'_{\mathcal{W},i} \subseteq N_1$ and $N_1 \neq N_\ell$.
4. $\alpha = R$ if $K'_{\mathcal{W},i} \subseteq N_\ell$ and $N_1 \neq N_\ell$.
5. $\alpha = LR$ if $K'_{\mathcal{W},i} \subseteq N_1$ and $N_1 = N_\ell$.

Thus, any p-subgraph $G'_{\mathcal{W},i}$ that satisfies Lemma 4.1 is characterized by a unique state-vector in $\mathcal{S}^i$. Given a function $f^i : \mathcal{G}_\mathcal{W} \rightarrow \mathcal{S}^i$ we now describe an algorithm to compute the function $\Theta(G_\mathcal{W}, f^i)$ which evaluates to 1 if and only if all members in $\mathcal{F}(G_\mathcal{W}, f^i)$ are operational.

Algorithm 4.1:
Input: a polygon diagram of G, a set K of terminals, a cast $\mathcal{W}$ and f^i as above.
Output: $\Theta(G_\mathcal{W}, f^i)$.

1. Construct the graph $R'_\mathcal{W}$ by adding the following set of new edges to the representative graph $R_\mathcal{W}$ (cf. Definition 5): $\{(x, y)|\ x\ and\ y\ represent\ extreme\ chords\ in\ the\ same\ window,\ say\ W_i,\ and\ f^i(G_{\mathcal{W},i}) = LR\}$. Define a *target* vertex in $R'_\mathcal{W}$ to be one that represents some extreme-left (extreme-right or left/right) set in a window W_i where $f^i(G_{\mathcal{W},i}) = L$ (respectively, $f^i(G_{\mathcal{W},i}) = R$ or $f^i(G_{\mathcal{W},i}) = LR$). Set $\Theta(G_\mathcal{W}, f^i) = 1$ if $R'_\mathcal{W}$ has no targets or if all possible targets are connected in the modified graph $R'_\mathcal{W}$. This step requires $O(t^2)$ time.
2. end.

Correctness of the above algorithm is straightforward. Given a cast $\mathcal{W}$ we outline in the next section an $O(n^4)$ algorithm to compute $\{Pr(G_{\mathcal{W},i}, \alpha)|\ \ G_{\mathcal{W},i} \in \mathcal{G}_\mathcal{W}\ and\ \ \alpha \in \mathcal{S}^i\}$(cf. Lemma 4.6). Assuming the correctness of this latter algorithm, we now proof the main theorem of this section.

Theorem 4.1 *Given a t-gon representation of G, $|V(G)| = n$, the UN–Rel_K reliability problem on G can be solved in $O(n^{O(t^2)})$ time.*

Proof: There are $O(n^4)$ windows in the given representation and each cast has at most $\binom{t}{2}$ windows. Hence, there are $O(n^{4\binom{t}{2}})$ casts. For every cast $\mathcal{W}$, the algorithm computes in one step $Pr(G_{\mathcal{W},i}, \alpha)$, for every $\alpha \in \mathcal{S}^i$ and $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$. Now, the total number of vertices in $\mathcal{G}_{\mathcal{W}} = \{G_{\mathcal{W},1}, \ldots, G_{\mathcal{W},|I|}\}$ is at most $n + 4|\mathcal{W}|$ ($\in O(n)$) (since each member $G_{\mathcal{W},i}$ has at most $|W_i| + 4$ vertices). By Lemma 4.6, the above probabilities can be computed in $O(n^4)$ time.

Now, $|\mathcal{S}^i| = 5$ and hence the total number of functions $f^i : \mathcal{G}_{\mathcal{W}} \rightarrow \mathcal{S}^i$ is $5^{\binom{t}{2}}$. Subsequently, the algorithm computes $Pr(G_{\mathcal{W}}, f^i)$, $\Theta(G_{\mathcal{W}}, f^i)$ and $Rel(G_{\mathcal{W}})$, for every such function f^i. For each f^i, the algorithm computes the above values in $O(t^2)$. Hence, the algorithm requires $O(n^4 + 5^{\binom{t}{2}} t^2)$ to compute $Rel(G_{\mathcal{W}})$ for each cast and consequently $O(n^{4\binom{t}{2}}(n^4 + 5^{\binom{t}{2}} t^2))$ time to process all casts. ∎

4.1 The K-terminal Reliability Problem on Permutation Graphs

Let $\mathcal{W}$ be a cast and $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$ be as in the previous section. Our objective is to compute $Pr(G_{\mathcal{W},i}, \alpha)$, $\alpha \in \mathcal{S}^i$. To this end, we define a set $\mathcal{S}^{ii}$ of state vectors (applicable to the p-subgraphs of $G_{\mathcal{W},i}$). Subsequently, we define the conditioned sets $(\mathcal{S}^{ii})_\alpha$, $\alpha \in \mathcal{S}^i$, required in equation 3. In the next section, we show how to compute the probabilities $Rel(G_{\mathcal{W},i}, \beta)$, $\beta \in \mathcal{S}^{ii}$. An essential ingredient in defining $\mathcal{S}^{ii}$ and proving some of its properties is the permutation $\pi(G_{\mathcal{W},i})$ defined below.

Definition 7. Consider the restriction of the polygon diagram of G to chords in $G_{\mathcal{W},i}$. Let d_x and d_y be the two sides defining W_i. For each chord $c \in Next(W_i)$ having no end-point on d_x (also d_y) extend c and d_x (respectively, d_y) until both line segments intersect each other. Denote the extended line segments d_x and d_y by d'_x and d'_y, respectively. The diagram of the line segments $W_i \bigcup Next(W_i)$ between d'_x and d'_y is a *permutation diagram* and hence we may view d'_x and d'_y as two horizontal lines with d'_x drawn above d'_y.

With this in mind, define the *left-side* (*right-side*) of the diagram to be the side where the end-point of d'_y is also an end-point of some chord in $ext_l(W_i) \bigcup Next_l(W_i)$ (respectively, $ext_r(W_i) \bigcup Next_r(W_i)$). Assume that $|V(G_{\mathcal{W},i})| = n$. We then assign the labels $1, 2, \ldots, n$ (in that order) to the chords whose end-points are encountered when traversing d'_y from left to right. Let $\pi(G_{\mathcal{W},i}) = (\pi_1, \pi_2, \ldots, \pi_n)$ be the permutation induced by the ordering of the end-points on d'_x when traversed from left to right. ∎

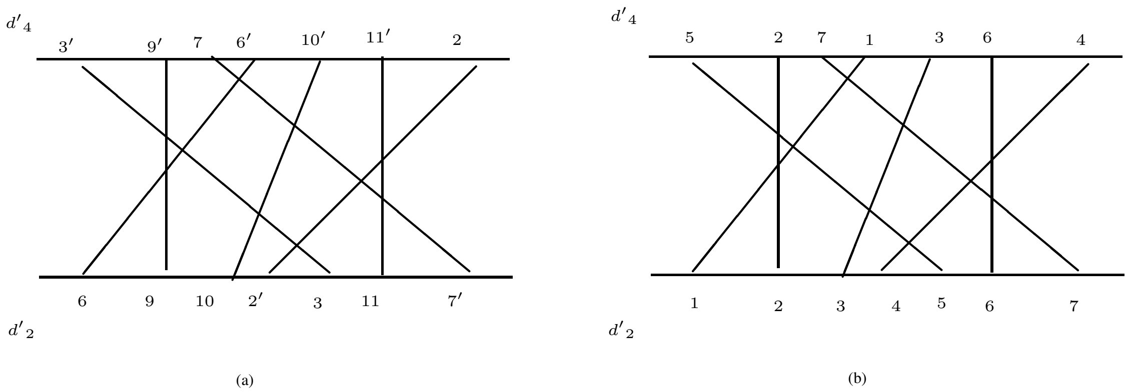


Figure 2

In our example (Figure 1), let $\mathcal{W} = \{C_1, \ldots, C_6\}$. Then $V(G_{\mathcal{W},5}) = \{2, 3, 6, 7, 9, 10, 11\}$. The permutation diagram defined above for $G_{\mathcal{W},5}$ is shown in Figure 2(a). Relabelling the vertices $1, 2, \ldots, etc.$ we get the diagram in Figure 2(b). Hence, $\pi(G_{\mathcal{W},5}) = (5, 2, 7, 1, 3, 6, 4)$. Note that, the set of vertices $\{1, \pi_1\} \in ext_l(W_i) \bigcup Next_l(W_i)$ according to the new labelling of the vertices . By symmetry, we also have $\{n, \pi_n\} \in ext_r(W_i) \bigcup Next_r(W_i)$

Let $H_{\mathcal{W},i}$ be the permutation graph defined by $\pi(G_{\mathcal{W},i})$. Before defining the set $\mathcal{S}^{ii}$, we introduce the following result used to prove some important properties of $\mathcal{S}^{ii}$.

Lemma 4.2 *The components of any p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ are identical to the components of the subgraph induced by $V(G'_{\mathcal{W},i})$ in $H_{\mathcal{W},i}$.*

Proof. It is easy to see that if $G_{\mathcal{W},i} \neq H_{\mathcal{W},i}$ then $H_{\mathcal{W},i}$ can be obtained from $G_{\mathcal{W},i}$ be adding at most 2 edges, each possible edge joins two vertices in $Next_l(W_i)$ or $Next_r(W_i)$. Furthermore, in $G_{\mathcal{W},i}$ the subgraphs induced by $ext_l(W_i) \bigcup Next_l(W_i)$ and $ext_r(W_i) \bigcup Next_r(W_i)$ are connected. The result then follows since $ext_l(W_i) \bigcup Next_l(W_i)$ and $ext_r(W_i) \bigcup Next_r(W_i)$ are assumed to be perfectly reliable in $G_{\mathcal{W},i}$ and $G'_{\mathcal{W},i}$. ■

Assuming that $V(G_{\mathcal{W},i})$ has been relabelled as in Definition 7, we now introduce the set $\mathcal{S}^{ii}$. Let $\{N_1, N_2, ..., N_\ell\}$ be the possible components of a given p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ where $ext_l(W_i) \bigcup Next_l(W_i) \subseteq N_1$ and $ext_r(W_i) \bigcup Next_r(W_i) \subseteq N_\ell$, as above. That is, the vertices $\{1, \pi_1\} \subseteq N_1$ and $\{n, \pi_n\} \subseteq N_\ell$. Define the *leader* of a component N_i to be the largest labelled vertex in that component. Roughly speaking, the $\mathcal{S}^{ii}$-state of $G'_{\mathcal{W},i}$ keeps information about the existence of operational terminals in two components: N_1 and a second component (denoted $N_{1+\varepsilon}$) that includes the first operational terminal in the sequence $\pi(G_{\mathcal{W},i})$ that is not included in N_1. Additional information is kept for the operational terminals that are not included in $N_1 \bigcup N_{1+\varepsilon}$.

More to the point, the state $\beta \in \mathcal{S}^{ii}$ of $G'_{\mathcal{W},i}$ is a 5-tuple (l_1, s, l_2, l_3, u), where

1. $\beta[l_1]$ is the leader of N_1.

2. $\beta[s] = 1$ if N_1 contains at least one operational terminal, otherwise, $\beta[s] = \perp$.

3. If N_1 does not contain all the operational terminals then set $\beta[l_2]$ to be the leader of a component $N_{1+\varepsilon}$ containing the first operational terminal π_i in the sequence $(\pi_1, \pi_2, ..., \pi_n)$ where $\pi_i \notin N_1$. Otherwise, $\beta[l_2] = \perp$.

4. $\beta[l_3]$ is the largest labelled operational vertex in $G'_{\mathcal{W},i}$.

5. If $G'_{\mathcal{W},i}$ has a set $K''_{\mathcal{W},i}$ of operational terminals that are not contained in $N_1 \bigcup N_{1+\varepsilon}$ then $\beta[u]$ is the smallest labelled operational terminal in $K''_{\mathcal{W},i}$. Otherwise, $\beta[u] = \perp$.

Using Lemma 4.2, we now draw some basic remarks on the state β of $G'_{\mathcal{W},i}$. First, if N_1, $N_{1+\varepsilon} \neq \phi$ then every vertex $x \in N_1$ that occurs before $\beta[l_2]$ in $\pi(G_{\mathcal{W},i})$ satisfies $x < \beta[l_2]$. Otherwise, $\beta[l_2] \in N_1$, contradicting the definition. In fact, all vertices in N_1 occur in $\pi(G_{\mathcal{W},i})$ before $\beta[l_2]$. To see this, assume (to derive a contradiction) that $y \in N_1$ and y occurs in $\pi(G_{\mathcal{W},i})$ after $\beta[l_2]$. Clearly, $y > \beta[l_2]$ otherwise $\beta[l_2] \in N_1$, a contradiction. Now, $y \in N_1$ if and only if there exist two vertices x and z, such that x, $\beta[l_2]$, y and z appear in $\pi(G_{\mathcal{W},i})$ in that order and x, $y > z$. It then follows from our first remark that $\beta[l_2] > x$. Hence, $\beta[l_2] > z$; consequently $\beta[l_2] \in N_1$, a contradiction. Thus, if $\beta[l_1]$, $\beta[l_2] \geq 1$ then $\beta[l_1] < \beta[l_2] \leq \beta[l_3]$. Finally, if $\beta[u] \geq 1$ then $G'_{\mathcal{W},i}$ violates the conditions in Lemma 4.1.

We now define $(\mathcal{S}^{ii})_\alpha$ for each $\alpha \in \mathcal{S}^i$:

1. $(\mathcal{S}^{ii})_\phi = \{\beta | \; \beta \in \mathcal{S}^{ii}, \beta[u] = \perp, \beta[s] = \perp \; and \; \beta[l_2] = \perp\}$

2. $(\mathcal{S}^{ii})_M = \{\beta | \; \beta \in \mathcal{S}^{ii}, \beta[u] = \perp, \beta[s] = \perp \; and \; \perp \neq \beta[l_2] \neq n\}$

3. $(\mathcal{S}^{ii})_L = \{\beta | \; \beta \in \mathcal{S}^{ii}, \beta[u] = \perp, \beta[s] = 1, \; \perp \neq \beta[l_1] \neq n \; and \; \beta[l_2] = \perp\}$

4. $(\mathcal{S}^{ii})_R = \{\beta | \; \beta \in \mathcal{S}^{ii}, \beta[u] = \perp, \beta[s] = \perp \; and \; \beta[l_2] = n\}$

5. $(\mathcal{S}^{ii})_{LR} = \{\beta | \; \beta \in \mathcal{S}^{ii}, \beta[u] = \perp, \beta[s] = 1 \; and \; \beta[l_1] = n\}$

Lemma 4.3 *Let $\mathcal{S}^{ii}$ and $(\mathcal{S}^{ii})_\alpha$, $\alpha \in \mathcal{S}^i$ be as above. Then equation 3 can be used to compute $Pr(G_{\mathcal{W},i}, \alpha)$, $\alpha \in \mathcal{S}^i$.*

Proof. By inspection of the above definitions, one may verify that: (1) any p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ is characterized by a unique vector in $\mathcal{S}^{ii}$, (2) if $\beta \in (\mathcal{S}^{ii})_\alpha$ then $\mathcal{F}(G_{\mathcal{W},i}, \beta) \subseteq \mathcal{F}(G_{\mathcal{W},i}, \alpha)$ and (3) the sets $(\mathcal{S}^{ii})_\alpha$, $\alpha \in \mathcal{S}^{ii}$, are disjoint. ∎

4.2 Computing the Refined Probabilities

A procedure for computing the list $L = (Pr(G_{\mathcal{W},i}, \beta) | \beta \in \mathcal{S}^{ii})$ for a given graph $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$ now follows. Let $\pi(G_{\mathcal{W},i})$ and $H_{\mathcal{W},i}$ be as above. The procedure is based on the following observation

Lemma 4.4 *For any state $\beta \in \mathcal{S}^{ii}$, $Pr(G_{\mathcal{W},i}, \beta) = Pr(H_{\mathcal{W},i}, \beta)$.*

Proof. The state β of a p-subgraph $G'_{\mathcal{W},i} \subseteq G_{\mathcal{W},i}$ is determined by the components of $G'_{\mathcal{W},i}$. By Lemma 4.2, $G'_{\mathcal{W},i} \in \mathcal{F}(G_{\mathcal{W},i}, \beta)$ if and only if $G'_{\mathcal{W},i} \in \mathcal{F}(H_{\mathcal{W},i}, \beta)$. ■

We henceforth use $H_{\mathcal{W},i}$ to compute $Pr(G_{\mathcal{W},i}, \beta)$, $\beta \in \mathcal{S}^{ii}$. For convenience, let H_j, $1 \leq j \leq n$, be the subgraph of $H_{\mathcal{W},i}$ induced by the set $\{\pi_1, \ldots, \pi_j\}$ of vertices and let H_0 denote the empty graph. In addition, denote by L_j, $0 \leq j \leq n$, the list $(Pr(H_j, \beta)|\beta \in \mathcal{S}^{ii})$ where β is a possible state of H_j. Thus, the required list L is just L_n. Viewing L_j as an array, we write $L_j[\beta]$ for $Pr(H_j, \beta)$.

The algorithm works iteratively be considering the vertices $\pi_1, \pi_2, \ldots, \pi_n$ in that order. Initially, L_0 has one state β whose 5 indices are initialized to the special value $\perp$ (cf. section 2). In addition, $L_0[\beta] = \perp$. Subsequently, at the jth step the algorithm computes L_j by including and excluding the vertex π_j to and from each entry in L_{j-1}.

Adding a vertex π_j to a subgraph of H_{j-1} in state β results in a larger subgraph in some state β'. We write $\beta' = \beta \oplus \pi_j$. The details of computing β' is given by Algorithm 4.2 below. The inclusion step sets $L_j[\beta \oplus \pi_i] = L_j[\beta \oplus \pi_i] + (L_{j-1}[\beta] \times p_{\pi_j})$. The exclusion step is performed if $p_{\pi_j} < 1$; here we set $L_j[\beta] = L_j[\beta] + (L_{j-1}[\beta] \times (1 - p_{\pi_j}))$. In the above assignment expressions, entries of L_j that appear in the right-hand side with no pre-assigned values are assumed to be equal to $\perp$.

Algorithm 4.2
Input: a state β of a p-subgraph $H' \subseteq H_{j-1}$ and a vertex π_j.
Output: the state $\beta' = \beta \oplus \pi_j$ of $H' \bigcup \pi_j$.

1. $\beta' = \beta$;
2. if ($\beta[l_1] = \perp$) then $\{\beta'[l_1] = \pi_j$; if ($\pi_j \in K$) then $\beta'[s] = 1\}$.
3. if ($\beta[l_1] \geq 1$ and $\beta[l_2] = \perp$) then
   (a) if $\pi_j < \beta[l_1]$ then
   $\{\beta'[l_2] = \beta'[u] = \perp$; $\beta'[l_1] = \beta[l_3]$; if ($\pi_j \in K$) then $\beta'[s] = 1\}$;
   (b) if ($\beta[l_1] < \pi_j$ and $\pi_j \in K$) then $\beta'[l_2] = max(\beta[l_3], \pi_j)$;
4. if ($\beta[l_1], \beta[l_2] \geq 1$) then
   (a) if ($\pi_j < \beta[l_1]$) then use rule (a) above;
   (b) if ($\beta[l_1] < \pi_j < \beta[l_2]$) then $\beta'[l_2] = \beta[l_3]$;
   (c) if ($\beta[l_2] < \pi_j$ and $\pi_j \in K$) then $\beta'[u] = min(\beta[u], \pi_j)$;
5. $\beta'[l_3] = max(\beta[l_3], \pi_j)$;
6. end.

Lemma 4.5 *Let β and H' be as specified in Algorithm 4.2. The algorithm computes the state β' of $H' \cup \pi_j$.*

Proof. By exhausting all possible cases. We first partition the state-vectors in $\mathcal{S}^{ii}$ into 3 classes based on the values of the indices l_1 and l_2 being (i) equal to $\perp$, (ii) $\beta[l_1] \geq 1$ and $\beta[l_2] = \perp$ and (iii) $\beta[l_1], \beta[l_2] \geq 1$. Following the assignment $\beta' = \beta$ in step 1, the algorithm assigns possible new values to certain indices in β' as required. Steps 2, 3 and 4 handle cases i, ii and iii, respectively. Now, assume that the components $N_1, N_2, ..., N_\ell$ of H' are labelled as above with the set of vertices $\{1, \pi_1\} \in N_1$ and $\{n, \pi_n\} \in N_\ell$. We show two sample cases; the remaining cases follow using a similar argument.

Case 3.b: By assumptions, $\beta[l_1] \geq 1$ (hence, $N_1 \neq \phi$) and $\beta[l_2] = \perp$ (hence, all operational terminals of H' belong to $N_1 \subseteq H'$). If $\beta[l_1] < \pi_j$ and $\pi_j \in K$ then π_j is the first operational terminal not in N_1. Denote the component of $H' \bigcup \pi_j$ that contains π_j by $N_{1+\varepsilon}$. We now verify the instructions in 3.b. If $\beta[l_3] > \pi_j$ then $\beta[l_3] \in N_{1+\varepsilon}$ (since $\beta[l_3]$ occurs before π_j in π). In any case, (i.e. whether $\beta[l_3] <$ or $> \pi_j$) the leader $\beta'[l_2]$ of $N_{1+\varepsilon}$ is $max(\beta[l_3], \pi_j)$. On the other hand, if $\beta[l_1] < \pi_j$ and $\pi_j \notin K$ the only required step is 5. Note that throughout case 3 whether $\pi_j \in K$ or not we have $\beta'[u] = \beta[u] = \perp$.

Case 4.c: By assumptions, $N_1, N_{1+\varepsilon} \neq \phi$ and $\beta[l_2] < \pi_j$. If $\pi_j \in K$ then π_j appears in $H' \cup \pi_j$ in a component different from N_1 and $N_{1+\varepsilon}$, hence the assignment $\beta'[u] = min(\beta[u], \pi_j)$. On the other hand, if $\pi_j \notin K$ then π_j can only affect the value of $\beta'[l_3]$, as in step 5. ■

Lemma 4.6 *Let $G_{\mathcal{W},i}$, $|V(G_{\mathcal{W},i})| = n$, and $\mathcal{S}^{ii}$ be as in this section. The above algorithm computes $Pr(H_{\mathcal{W},i}, \beta) = Pr(G_{\mathcal{W},i}, \beta)$, $\beta \in \mathcal{S}^{ii}$, in $O(n^4)$ time. Hence, $\{Pr(G_{\mathcal{W},i}, \alpha) | \ \alpha \in \mathcal{S}^i\}$ can be computed in $O(n^4)$ time.*

Proof.
Correctness: It suffices to show that L_j, $j \geq 1$, is *complete* (that is, L_j contains an entry for each possible state $\beta \in \mathcal{S}^{ii}$ of H_j) and *correct* given that L_{j-1} is complete and correct (with respect to H_{j-1}). The completeness part follows since the algorithm computes L_j by performing inclusion and exclusion steps of π_j on each entry in L_{j-1}. Hence, exhausting all the possibilities. The correctness part follows from Lemma 4.5.

Timing: The total number of distinct states in $\mathcal{S}^{ii}$ is $O(n^3)$ and hence the number of entries in each list L_j, $1 \leq j \leq n$, is at most $O(n^3)$. Furthermore, in constructing a list L_j from a previous list L_{j-1} the algorithm processes each entry in L_{j-1} in $O(1)$ time. Thus, $Pr(G_{\mathcal{W},i}, \beta)$, $\beta \in \mathcal{S}^{ii}$, can be computed in $O(n^4)$ time.
Finally, it is easy to that $\{Pr(G_{\mathcal{W},i}, \alpha) \mid \alpha \in \mathcal{S}^i\}$ can be computed simultaneously with L_n. ■

5 The K-Resource Reliability Problem

As in section 4, let $\mathcal{W}$ be a cast and $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$. Denote the set of terminals (service centers) in $G_{\mathcal{W},i}$ by $K_{\mathcal{W},i}$. In addition, let $G'_{\mathcal{W},i}$ be a p-subgraph of $G_{\mathcal{W},i}$ with components labelled $N_1, N_2, \ldots, N_\ell$ where $ext_l(W_i) \bigcup Next_l(W_i) \subseteq N_1$ and $ext_r(W_i) \bigcup Next_r(W_i) \subseteq N_\ell$, as before. Furthermore, denote the operational terminals in $G'_{\mathcal{W},i}$ by $K'_{\mathcal{W},i}$. The following lemma is analogous to Lemma 4.1. The proof is straightforward and hence omitted.

Lemma 5.1 *$G'_{\mathcal{W},i}$ is a candidate to appear in some operational subnetwork of G if every possible nonempty (internal) component N_i, $i \neq 1, \ell$, of $G'_{\mathcal{W},i}$ has at least one operational terminal. That is, it is possible that neither N_1 nor N_ℓ contains an operational terminal of $G'_{\mathcal{W},i}$.*

Define $\mathcal{S}^i$ to be the set $\{A_0, A_1, D_0, D_L, D_R, D_{LR}\}$. We use A_0 and A_1 to characterize p-subgraphs with $N_1 = N_\ell$. The remaining states (i.e. D_0, D_L, D_R and D_{LR}) are used to characterize p-subgraphs with $N_1 \neq N_\ell$. For the purpose of defining the exact state α of $G'_{\mathcal{W},i}$ we introduce the new variables $T_1, T_\ell \in \{true, false\}$, $a \in \{0,1\}$ and and $b \in \{0, L, R, LR\}$. Set $T_1 = true$ ($T_\ell = true$) if N_1 (respectively, N_ℓ) contains an operational terminal. Otherwise, $T_1 = false$ ($T_\ell = flase$). Assign values to a and b as follows. If $\overline{T_1}\ \overline{T_2}$ then $a = b = 0$ else $a = 1$. In addition, if $T_1\overline{T_2}$ (by symmetry, $\overline{T_1}T_2$) then $b = L$ (respectively, $b = R$). Finally, if T_1T_2 then $b = LR$. We now define α as follows:

1. If $N_1 = N_\ell$ then $\alpha = A_a$.
2. Otherwise, if $N_1 \neq N_\ell$ and every component N_i, $1 < i < \ell$ contains a terminal then $\alpha = D_b$.

A straightforward algorithm to compute $\Theta(G_{\mathcal{W}}, f^i)$ for a given function $f^i : \mathcal{G}_{\mathcal{W}} \rightarrow \mathcal{S}^i$ now follows.

Algorithm 5.1:
Input: a polygon diagram of G, a set K of terminals, a cast $\mathcal{W}$ and f^i as above.
Output: $\Theta(G_{\mathcal{W}}, f^i)$.

1. Construct the graph $R'_{\mathcal{W}}$ by adding the following set of new edges to the representative graph $R_{\mathcal{W}}$ (cf. Definition 5): $\{(x,y)|\ x$ and y represent chords in the same window, say W_i, where $f^i(G_{\mathcal{W},i}) = A_0$ or $A_1\}$. Using the information encoded in $f^i(G_{\mathcal{W},i})$, $G_{\mathcal{W},i} \in \mathcal{G}_{\mathcal{W}}$, identify the components of $R'_{\mathcal{W}}$ that contain terminals.
2. Set $\Theta(G_{\mathcal{W}}, f^i) = 1$ if $R'_{\mathcal{W}} = \phi$ or if $R'_{\mathcal{W}} \neq \phi$ and every component in $R'_{\mathcal{W}}$ contains a terminal.
3. end.

The above algorithm requires $O(t^2)$ time. As will be shown in Lemma 5.2, the set $\{Pr(G_{\mathcal{W},i}, \alpha)|$ $\alpha \in \mathcal{S}^i\}$ can be computed in $O(n^{k+4})$ where $k = |K|$. We then have

Theorem 5.1 *Given a t-gon representation of G, $|V(G)| = n$, the K-resource reliability problem on G can be solved in $O(n^{O(t^2+k)})$.*

Proof: By Lemma 5.2 and the structure of $\mathcal{S}^i$ one can deduce the following time bound $O(n^{4\binom{t}{2}}\ (n^{k+4} + 6^{\binom{t}{2}}t^2))$. ■

5.1 The K-Resource Reliability Problem on Permutation Graphs

For convenience, let $k_i = |K_{\mathcal{W},i}|$ throughout this section. In addition, assume that $V(G_{\mathcal{W},i})$ has been relabelled as in Definition 7. If $G'_{\mathcal{W},i}$ satisfies Lemma 5.1 then it has at most $k_i + 2$ components. Each state in $\mathcal{S}^{ii}$ is a vector in $\{\perp, 1, 2, \ldots, |V(G_{\mathcal{W},i})|\}^{2(k_i+3)}$. Briefly, the $\mathcal{S}^{ii}$-state of a p-subgraph $G'_{\mathcal{W},i}$ keeps track of the existence of operational service centers in the first possible $k_i + 2$ components (assuming that we start from the component N_1 and scan the vertices from left to right according to the permutation $\pi(G_{\mathcal{W},i})$). We require a total of $2k_i + 4$ numbers to encode this information. The existence of operational service centers in the remaining possible components of $G'_{\mathcal{W},i}$ is encoded (in less detail) by two additional numbers in the state-vector. We now describe the structure of any state-vector in detail. The coordinates of each vector are denoted $(l_1, r_1, l_2, r_2, \ldots, l_{k_i+3}, r_{k_i+3})$. If $G'_{\mathcal{W},i}$ has $\ell \geq 0$ components then define its state β as follows.

1. $\beta[l_i]$, $1 \leq i \leq min(\ell, k_i + 2)$, is the leader of component N_i. For all other possible values of i in the range: $min(\ell, k_i + 2) < i \leq k_i + 2$, $\beta[l_i] = \perp$. In addition, if $\ell > k_i + 2$ then $\beta[l_{k_i+3}]$ is the largest labelled vertex in $\{N_{k_i+3}, \ldots, N_\ell\}$; otherwise, $\beta[l_{k_i+3}] = \perp$.

2. For $1 \leq i \leq k_i + 2$, $\beta[r_i] = 1$ if N_i contains a terminal; otherwise, $\beta[r_i] = \perp$. Furthermore, if $\ell > k_i + 2$ then $\beta[r_{k_i+3}] = 1$ if any component in $\{N_{k_i+3}, \ldots, N_\ell\}$ contains a terminal. Otherwise, $\beta[r_{k_i+3}] = \perp$.

Conversely, if $\beta \in \mathcal{S}^{ii}$ then any p-graph $G'_{\mathcal{W},i}$ characterized by β has the following properties. (We assume that the possible components $N_1, \ldots, N_\ell$ of $G'_{\mathcal{W},i}$ are labelled as above. In addition, we use T_1 and T_ℓ as in the definition of $\mathcal{S}^i$.)

1. $\ell \leq k_i + 2$ if and only if $\beta[l_{k_i+3}] = \perp$.

2. If $\ell \leq k_i + 2$ then every possible component N_i, $1 < i < k_i + 2$, has an operational service center if and only if $\beta[r_i] = 1$.

3. $N_1 = N_\ell$ if and only if $\beta[l_1] = n$.

4. $T_1 = true$ ($T_\ell = true$) if and only if $\beta[r_1] = 1$ (respectively, $\beta[r_\ell] = 1$).

Using the above observations, one can readily compute each set $(\mathcal{S}^{ii})_\alpha$, $\alpha \in \mathcal{S}^i$, required in equation 3.

5.2 Computing The Refined Probabilities

Let $\pi(G_{\mathcal{W},i})$ and $H_{\mathcal{W},i}$ be as in section 4.1. One may then verify that Lemma 4.4 holds for any state $\beta \in \mathcal{S}^{ii}$. Hence, it suffices to compute $\{Pr(H_{\mathcal{W},i}, \beta) | \ \beta \in \mathcal{S}^{ii})\}$. We use the inclusion/exclusion algorithm presented in section 4.2 combined with Algorithm 5.2 shown below.

Algorithm 5.2:

Input: a state β of a p-subgraph $H' \subseteq H_{j-1}$ and a vertex π_j.
Output: the state $\beta' = \beta \oplus \pi_j$ of $H' \bigcup \pi_j$.
Notation: Recall that $k_i = |K_{\mathcal{W},i}|$. Let ℓ be the largest index such that $\beta[l_\ell] > 1$, otherwise, $\ell = \perp$. In addition, let x be the smallest index such that $\pi_j < \beta[l_x]$. If no such x exists then $x = \perp$. Also, let $term(\pi_j) = 1$ if $\pi_j \in K$, otherwise, $term(\pi_j) = \perp$.

1. $\beta' = \beta$;

2. if ($\ell > 1$ *and* $\pi_j < \beta[l_\ell]$) then {

   $\beta'[l_x] = max(\beta[l_\ell], \beta[l_{k_i+3}])$;
   $\beta'[r_x] = \beta[r_x] \vee \beta[r_{x+1}] \vee \ldots \vee \beta[r_\ell] \vee term(\pi_j)$;
   if ($x \leq k + 2$) then $\beta'[l_{x+1}, \ldots, r_{k_i+3}] = \perp$};

3. if ($\ell = \perp$ *or* $\pi_j > \beta[l_\ell]$) then {

$$\beta'[l_{min(\ell+1,k_i+3)}] = \pi_j;$$
$$\beta'[r_{min(\ell+1,k_i+3)}] = term(\pi_j); \};$$

4. end.

As in Lemma 4.5, one can proof the correctness of the above algorithm by exhausting all possible cases. Here, any state vector in $\mathcal{S}^{ii}$ satisfies either the conditions in step 2 or those in step 3. If the conditions in step 2 hold then adding π_j to H' joins the components N_x, N_{x+1}, ..., N_ℓ into a new component denoted N_x in $H' \bigcup \pi_j$. This new component will have an operational service center if either $\pi_j \in K$ or one of its subcomponents include an operational service center. On the other hand, the conditions in step 3 are satisfied if either H' is empty (the case where $\ell = \perp$) or $H' \bigcup \pi_j$ has one more component than H' (this occurs if $\pi_j > \beta[l_\ell]$). In either case, verification of the corresponding executable statements is straightforward. We then have the following

Lemma 5.2 *The set* $\{Pr(G_{\mathcal{W},i}, \alpha) |\ \alpha \in \mathcal{S}^i\}$ *can be computed in* $O(n^{k+4})$ *time.*

6 Concluding Remarks

In this paper we outlined two polynomial time algorithms to solve the $UN\text{–}Rel_K$ problem and the K-resource reliability problem on t-polygon graphs, for fixed t. An approach for solving a number of optimization problems on t-polygon graphs also appears in [6]. The applicability of the above algorithms relies on the availability of a polygon representation of the input graphs. Recognizing t-polygon graphs, for fixed $t \geq 3$, seems to be an open problem that warrants further research. Nevertheless, given a graph G one can use the circle representation produced by the algorithm of [7] to find an upper bound on the minimum t such that G is a t-polygon graph.

Acknowledgements

Research of the author is supported by NSERC Canada under grant numbers OGP36899.

References

[1] H. M. AboElFotoh and C. J. Colbourn, *Efficient algorithms for computing the reliability of permutation and interval graphs*, Networks, 20 (1990), pp. 883–898.

[2] M. O. Ball, *Complexity of network reliability computations*, Networks, 10 (1980), pp. 153–165.

[3] M. O. Ball and J. S. Provan, *Bounds on the reliability polynomial for shellable independence systems*, SIAM J. Alg. Disc. Meth., 3 (1982), pp. 166–181.

[4] C. J. Colbourn, *The Combinatorics of Network Reliability*, Oxford University Press, 1987.

[5] C. J. Colbourn, A. Satyanarayana, C. Suffel, and K. Sutner, *Computing residual connectedness reliability for restricted networks*. To appear in Disc. Appl. Math.

[6] E. S. ELMALLAH AND L. K. STEWART, *Independence and domination in polygon graphs*. To appear in Disc. Math.

[7] C. P. GABOR, W. HSU, AND K. J. SUPOWIT, *Recognizing circle graphs in polynomial time*, in the 26th IEEE Symp. on Foundations of Computer Science, 1985, pp. 106–116.

[8] J. S. PROVAN, *The complexity of reliability computations on planar and acyclic graphs*, SIAM J. Comput., 15 (1986), pp. 694–702.

[9] J. S. PROVAN AND M. O. BALL, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Comput., 12 (1983), pp. 777–788.

[10] K. SUTNER, A. SATYANARAYANA, AND C. SUFFEL, *The complexity of the residual node connectedness reliability problem*, SIAM J. Computing, 20 (1991), pp. 149–155.

[11] L. G. VALIANT, *The complexity of enumeration and reliability problems*, SIAM J. Computing, 8 (1979), pp. 410–421.