Computing 2-Terminal Reliability of Probe Interval Graphs

Chao-Chun Ting and Min-Sheng Lin

Department of Electrical Engineering
National Taipei University of Technology
Taipei 106, Taiwan

Copyright © 2014 Chao-Chun Ting and Min-Sheng Lin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Consider a probabilistic graph G in which the edges are perfectly reliable, but vertices may fail with some known probabilities. The 2-terminal reliability of G is defined as the probability that one operational path exists between a given source and destination pair vertices of G . This 2-terminal reliability problem is known to be #P-complete for general graphs but solvable in polynomial time for interval graphs. This work presents a polynomial-time algorithm for computing the 2-terminal reliability of probe interval graphs, which is a superclass of interval graphs.

Keywords: Algorithms, 2-terminal reliability, Probe interval graphs

1 Introduction

A graph is an *interval graph* [4] if its vertices can be placed in a one-to-one correspondence with a family of intervals on the real line such that two vertices are adjacent if and only if their corresponding intervals have a non-empty intersection. A *probe interval graph* (PIG) [9,11] is a generalization of an interval graph in which the vertex set is partitioned into probes and non-probes, such that two vertices are adjacent if and only if their corresponding intervals intersect and at least one of them is a probe. This definition generalizes interval graphs, which correspond to cases in which all vertices are probes. Zhang [11] introduced the class of PIGs to model physical mapping of DNA sequences. Notably, PIGs are weakly chordal [9], but not chordal because they contain the graph C_4 that has no chord.

Many common graph problems on PIGs, such as those that involve dominating sets [5] or independent sets [10], are known to be solvable in polynomial time. This paper will investigate the problem of 2-terminal reliability on PIGs. Reliability analysis problems arise in many areas of engineering. A network can be modeled as a probabilistic graph, in which each edge and vertex has an associated probability of failure. This paper concerns the model in which the edges of the probabilistic graph are assumed to be perfectly reliable, whereas the vertices fail independently with known probabilities. For a probabilistic graph G and specified vertices s (source) and t (destination) of G , the *2-terminal reliability* (2TR) [1] is the probability that an operational path (a path of operational vertices) exists between s and t . The *K-terminal reliability* of a probabilistic graph is another related problem. For a probabilistic graph G and a distinguished set of target vertices K , the *K-terminal reliability* (KTR) [2, 5, 6] is the probability that operational paths exist between every pair of vertices in K . The 2TR problem is the special case with $K=\{s, t\}$.

AboElFotouh and Colbourn [1] confirmed that the problem of computing 2TR is #P-complete for general graphs and even for grid graphs. AboElFotouh and Colbourn [2] have also proposed polynomial-time algorithms for computing the 2TR of chordal graphs and their subclass of interval graphs. Lin [6] proposed a linear-time algorithm for computing the KTR (including the 2TR) of proper interval graphs, which is a subclass of interval graphs in which no interval is allowed to contain another one. The present authors [7] previously developed polynomial-time algorithms for computing the KTR (including the 2TR) of d -trapezoid graphs, which is another superclass of interval classes. This paper extends the results of that work [7] to computing the 2TR of PIGs.

2. Preliminaries

This section presents the preliminaries on which the desired algorithm depends.

Definition 1.

Let G be a graph and $V(G)$ be partitioned into probes and non-probes. Then G is a PIG if a set $\mathbb{I}$ of intervals exists on the real line such that each vertex $v \in V(G)$ corresponds to an interval $I_v \in \mathbb{I}$ and an edge $(u, v) \in E(G)$ if and only if intervals I_u and I_v intersect and at least one of u, v is a probe. The above geometric representation is called a *probe interval representation*, which is denoted by $D(G)$.

Let $a(I_v)$ and $b(I_v)$ represent the left and right endpoints of interval I_v on the real line, respectively. Without loss of generality, assume that no two intervals have a common endpoint. The endpoints of all intervals are labeled with distinct integers between 1 and $2n$ (inclusive), where n is the number of vertices in the PIG.

Consider a PIG G with n vertices and its probe interval representation $D(G)$. The following definitions are introduced.

Definition 2.

A scanline $s(x)$ of $D(G)$ is the vertical line at position $x \in \{0.5, 1.5, \dots, 2n+0.5\}$ of the real line in $D(G)$.

Definition 3.

Let $s(x)$ and $s(y)$ be two scanlines in $D(G)$ with $x \leq y$. Define a span $P(x, y)$ that consists of all intervals I_v for which

- (i) I_v is a probe and crosses at least one of $s(x)$ and $s(y)$, or has both endpoints between $s(x)$ and $s(y)$, and
- (ii) I_v is a non-probe and crosses both $s(x)$ and $s(y)$.

Notably, if intervals I_u and I_w lie on the left side of $s(x)$ and the right side of $s(y)$ respectively, then the removal of all vertices v such that $I_v \in P(x, y)$ clearly separates vertices u and w in distinct connected components [3].

Definition 4.

Let $P(x_1, y_1)$ and $P(x_2, y_2)$ be two distinct spans. The $P(x_1, y_1)$ is *less than* $P(x_2, y_2)$, denoted by $P(x_1, y_1) \prec P(x_2, y_2)$, if and only if $x_1 \leq x_2$ and $y_1 \leq y_2$.

Let $\Phi(G) = \{P(x, y) \mid x, y \in \{0.5, 1.5, \dots, 2n+0.5\}\}$ be the set of all spans in $D(G)$. According to the above definition, $\langle \Phi(G), \prec \rangle$ is a poset. Theorem 1 states that $\langle \Phi(G), \prec \rangle$ is also a bounded lattice, as can be easily confirmed using Definitions 3 and 4.

Theorem 1.

The poset $\langle \Phi(G), \prec \rangle$ is a bounded lattice with bottom element $P(0.5, 0.5)$ and top element $P(2n+0.5, 2n+0.5)$. If $P(x_1, y_1)$ and $P(x_2, y_2) \in \Phi(G)$, then the meet $P(x_1, y_1) \wedge P(x_2, y_2) = P(\min\{x_1, x_2\}, \min\{y_1, y_2\})$ and the join $P(x_1, y_1) \vee P(x_2, y_2) = P(\max\{x_1, x_2\}, \max\{y_1, y_2\})$.

The following properties will be used to prove Theorem 2 in the next section.

Property 1.

For $P_i, P_j \in \Phi(G)$, $(P_i \wedge P_j) \setminus P_j \subseteq P_i \setminus P_j$.

Proof. Consider the following three cases.

- (i) $P_i \prec P_j$. Since $P_i \wedge P_j = P_i$, the proof is trivial.
- (ii) $P_j \prec P_i$. Since $P_i \wedge P_j = P_j$, $(P_i \wedge P_j) \setminus P_j = \emptyset$, and therefore $(P_i \wedge P_j) \setminus P_j \subseteq P_i \setminus P_j$.
- (iii) Neither $P_i \prec P_j$ nor $P_j \prec P_i$. Let $P_i = P(x_i, y_i)$ and $P_j = P(x_j, y_j)$. Without loss of generality, assume $x_i < x_j < y_j < y_i$. Hence $P_i \wedge P_j = P(\min\{x_i, x_j\}, \min\{y_i, y_j\}) = P(x_i, y_j)$. If $I_v \in P(x_i, y_j)$ and it is a probe that satisfies condition (i) of Definition 3, then I_v is also in $P(x_i, y_i)$. However, if $I_v \in P(x_i, y_j)$ and it is a non-probe that satisfies condition (ii) of Definition 3, then I_v is also in $P(x_j, y_j)$. Therefore, $P_i \wedge P_j \subseteq P_i \cup P_j$ and $(P_i \wedge P_j) \setminus P_j \subseteq P_i \setminus P_j$. $\square$

Property 2.

For $P_i, P_j, P_k \in \Phi(G)$, if $P_i \prec P_j \prec P_k$, then $(P_i \setminus P_k) \setminus P_j = P_i \setminus P_j$.

Proof. Clearly, if $P_i \cap P_k \subseteq P_j$, $(P_i \setminus P_k) \setminus P_j = P_i \setminus P_j$. Therefore, the current task is to show that given $P_i \prec P_j \prec P_k$, $P_i \cap P_k \subseteq P_j$. Let $P_i = P(x_i, y_i)$, $P_j = P(x_j, y_j)$ and $P_k = P(x_k, y_k)$. First, assume $I_v \in P_i \cap P_k$ and I_v is a non-probe. Since $P_i \prec P_k$, I_v crosses both scanlines $s(x_i)$ and $s(y_k)$. Additionally, since $P_i \prec P_j \prec P_k$, $x_i < x_j < y_j < y_k$, and therefore I_v crosses both scanlines $s(x_j)$ and $s(y_j)$. Hence, $I_v \in P_j$. Next, assume $I_v \in P_i \cap P_k$ and I_v is a probe. Since $P_i \prec P_j \prec P_k$, five cases must be considered.

- (i) $x_i < x_j < y_i < y_j < x_k < y_k$. In this case, I_v must cross scanline $s(y_j)$ and hence $I_v \in P_j$.
- (ii) $x_i < y_i < x_j < x_k < y_j < y_k$. In this case, I_v must cross scanline $s(x_j)$ and hence $I_v \in P_j$.
- (iii) $x_i < y_i < x_j < y_j < x_k < y_k$. In this case, I_v must cross both scanlines $s(x_j)$ and $s(y_j)$, and hence $I_v \in P_j$.
- (iv) $x_i < x_j < y_i < x_k < y_j < y_k$. In this case, I_v crosses at least one of the scanlines $s(x_j)$ and $s(y_j)$, or has both endpoints between $s(x_j)$ and $s(y_j)$. Therefore, $I_v \in P_j$.
- (v) $x_i < x_j < x_k < y_i < y_j < y_k$. The argument is similar to that made in case (iv).

In all cases, if $I_v \in P_i \cap P_k$ and I_v is a probe, then $I_v \in P_j$. Therefore, $P_i \cap P_k \subseteq P_j$. $\square$

3. Polynomial-time algorithm for computing the 2TR of a FIG

This section presents a polynomial-time algorithm for computing the 2TR of a FIG. Consider a FIG G and its corresponding probe interval representation $D(G)$. Let s and t be the source vertex and the destination vertex respectively. Without loss of generality, assume that the left endpoint of I_s is less than the left endpoint of I_t ($a(I_s) < a(I_t)$) in $D(G)$. Obviously, all vertices v whose corresponding intervals I_v are on the left side of I_s ($b(I_v) < a(I_s)$) or on the right side of I_t ($a(I_v) > b(I_t)$) are irrelevant and can be eliminated without affecting the 2TR of G . Let $G_{s,t}$ denote the FIG that is obtained by eliminating these irrelevant vertices from G .

Let a_{min} be the minimum of $a(I_v)$ of all intervals I_v in $D(G_{s,t})$ and b_{max} be the maximum of $b(I_v)$ of all intervals I_v in $D(G_{s,t})$.

Definition 5.

Let $\Phi(G_{s,t})$ denote the set of all spans between the leftmost span and the rightmost span (inclusive). According to the definition of span, $\Phi(G_{s,t}) = \{P(x, y) \mid a_{min} - 0.5 \leq x \leq a(I_t) - 0.5 \text{ and } \max\{x, b(I_s) + 0.5\} \leq y \leq b_{max} + 0.5\}$.

Let each I_v fail randomly with probability q_v that equals the probability of the failure of vertex v . Therefore, the 2TR of G , denoted by $R(G_{s,t})$, is given by

$$R(G_{s,t}) = 1 - \Pr\left[\bigcup_{P \in \Phi(G_{s,t})} \{\text{all } I \in P \text{ fail}\}\right]. \quad (1)$$

Clearly, by Theorem 1, the poset $\langle \Phi(G_{s,t}), \prec \rangle$ is also a bounded lattice. To compute Eq. (1), for all spans $P_k \in \Phi(G_{s,t})$, consider the following event.

$E(P_k) = \{ \text{there exists an operational path that intersects all spans } P_i \in \Phi(G_{s,t}) \text{ with } P_i \prec P_k, \text{ but does not intersect } P_k \}.$

To simplify the implementation of the algorithm, a dummy span $P_\emptyset = P_{(2n+1, 2n+1)} = \emptyset$ is placed to the right of all spans of $\Phi(G_{s,t})$. Thus, $P \prec P_\emptyset$ for all $P \in \Phi(G_{s,t})$. Without loss of generality, all spans of $\Phi(G_{s,t}) \cup \{P_\emptyset\}$ are reordered in total order according to the poset $\langle \Phi(G_{s,t}) \cup P_\emptyset, \prec \rangle$. Let $\{P_1, P_2, \dots, P_m\}$ be the results of sorting, where $m = |\Phi(G_{s,t})| + 1$. Clearly, P_m is the dummy span $P_\emptyset$.

The occurrence of event $E(P_m)$ implies the existence of an operational path that intersects all spans of $\Phi(G_{s,t})$, so intervals I_s and I_t can be connected by this operational path. Therefore, $\Pr[E(P_m)]$ is just the probability that Eq. (1) holds.

The following theorem provides a dynamic programming method for computing $\Pr[E(P_m)]$ and it varies slightly from a theorem that was presented in our earlier work [7].

Theorem 2. $\Pr[E(P_k)] = 1 - \sum_{P_i \prec P_k} \left(\Pr[E(P_i)] \cdot \prod_{I_v \in P_i \setminus P_k} q_v \right).$ (2)

Proof. Based on Properties 1 and 2, Theorem 2 follows from Theorem 3 in our earlier work [7]. $\square$

Using Theorem 2, $\Pr[E(P_k)]$ can be computed in the order of $k=1, 2, \dots, m$, and our earlier work [7] proved that this computation can be performed in $O(n \cdot m^2)$ time, where $m = |\Phi(G_{s,t})| + 1$ and n is the number of vertices of G . Since the number of spans, m , of $D(G)$ is at most $O(n^2)$, the 2TR of an n -vertex FIG can be computed in $O(n^5)$ time, which can nevertheless be reduced to $O(n^4)$ as follows.

Let $F(i, k)$ denote the event that all intervals in $P_i \setminus P_k$ fail. Thus, $\Pr[F(i, k)] = \prod_{I_v \in P_i \setminus P_k} q_v$ and Eq. (2) can be rewritten as

$$\Pr[E(P_k)] = 1 - \sum_{P_i \prec P_k} \left(\Pr[E(P_i)] \cdot \Pr[F(i, k)] \right). \quad (3)$$

The following property yields a recursive characterization of the computation of $\Pr[F(i, k)]$.

Property 3. For $P_i, P_j, P_k \in \Phi(G_{s,t})$ and $P_i \prec P_j \prec P_k$,

$$\Pr[F(i, k)] = \Pr[F(i, j)] \cdot \prod_{I_v \in (P_j \setminus P_k) \cap (P_i)} q_v.$$

Proof. All intervals of $P_i \setminus P_k$ can be divided into two disjoint sets based on whether they belong to P_j . Thus, $P_i \setminus P_k = ((P_i \setminus P_k) \setminus P_j) \sqcup ((P_i \setminus P_k) \cap P_j)$. First, by Property 2, $(P_i \setminus P_k) \setminus P_j$ can be simplified to $P_i \setminus P_j$. Next, $(P_i \setminus P_k) \cap P_j$ can be rewritten as $(P_j \setminus P_k) \cap P_i$. Therefore, $P_i \setminus P_k = (P_i \setminus P_j) \sqcup ((P_j \setminus P_k) \cap P_i)$ and $\Pr[F(i, k)] = \Pr[F(i, j)] \cdot \prod_{I_v \in (P_j \setminus P_k) \cap (P_i)} q_v$. $\square$

Definition 6.

Span P' is said to be an *immediate predecessor* of span P if and only if $P' \prec P$ and no span P'' exists such that $P' \prec P'' \prec P$.

Let $Y(P)$ denote the set of all immediate predecessors of span P . The following property can be easily verified.

Property 4.

$$Y(P(x,y)) = \{P(x-1,y) \mid P(x-1,y) \in \Phi(G_{s,t})\} \cup \{P(x,y-1) \mid P(x,y-1) \in \Phi(G_{s,t})\}.$$

The bottleneck in the computation of $\Pr[E(P_k)]$ in Eq. (3) is the computation of $\Pr[F(i,k)]$ for all $P_i \prec P_k$. The method for accelerating the computation of $\Pr[F(i,k)]$ is based on the trivial observation that

$$\{P_i \mid P_i \prec P_k\} = Y(P_k) \cup \bigcup_{P_j \in Y(P_k)} \{P_i \mid P_i \prec P_j\}.$$

First, if $P_i \in Y(P_k)$, then $\Pr[F(i,k)]$ can be directly computed as

$$\Pr[F(i,k)] = \prod_{I_v \in (P_i \setminus P_k)} q_v. \quad (4)$$

Next, if $P_i \notin Y(P_k)$, then let P_j be an immediate predecessor of P_k such that $P_i \prec P_j \prec P_k$. Now, by Property 3, $\Pr[F(i,k)]$ can be indirectly computed as

$$\Pr[F(i,k)] = \Pr[F(i,j)] \cdot \prod_{I_v \in (P_j \setminus P_k) \cap (P_i)} q_v. \quad (5)$$

Based on the above formulations, the formal algorithm for computing the 2TR of a PIG is presented as below.

Algorithm 1. *Compute the 2TR of a PIG*

Input: A PIG G with its probe interval representation $D(G)$, a source vertex s and a destination vertex t .

Output: $R(G_{s,t}) \leftarrow$ the 2TR of G .

```

1  Step 1:// compute  $\Phi(G_{s,t})$  and  $Y(P)$  for each  $P \in \Phi(G_{s,t})$  //
2   $G_{s,t} \leftarrow$  the PIG obtained after eliminating the irrelevant vertices from  $G$ ;
3   $a_{min} \leftarrow \min_{I_v \in D(G_{s,t})} \{a(I_v)\}$ ;  $b_{max} \leftarrow \max_{I_v \in D(G_{s,t})} \{b(I_v)\}$ ;  $\Phi(G_{s,t}) \leftarrow \emptyset$ ;
4  for  $x \leftarrow a_{min}-0.5$  to  $a(I_t)-0.5$  step 1 do
5    for  $y \leftarrow \max\{x, b(I_s)+0.5\}$  to  $b_{max}+0.5$  step 1 do
6       $P(x,y) \leftarrow \emptyset$ ;
7      for each  $I_v$  of  $D(G_{s,t})$  do // Definition 3 //
8        if ( $I_v$  is a probe and not( $b(I_v) < x$  or  $a(I_v) > y$ )) then  $P(x,y) \leftarrow P(x,y) \cup \{I_v\}$ ;
9        if ( $I_v$  is a non-probe and  $a(I_v) < x$  and  $b(I_v) > y$ ) then  $P(x,y) \leftarrow P(x,y) \cup \{I_v\}$ ;
10     end-for
11      $\Phi(G_{s,t}) \leftarrow \Phi(G_{s,t}) \cup P(x,y)$ ;
12      $Y(P(x,y)) \leftarrow \emptyset$ ;
```

```

13   if  $(P(x-1,y) \in \Phi(G_{s,t}))$  then  $Y(P(x,y)) \leftarrow Y(P(x,y)) \cup P(x-1,y)$ ; //Property 4//
14   if  $(P(x,y-1) \in \Phi(G_{s,t}))$  then  $Y(P(x,y)) \leftarrow Y(P(x,y)) \cup P(x,y-1)$ ; //Property 4//
15   end-for
16 end-for
17  $P(2n+1, 2n+1) \leftarrow \emptyset$ ; // a dummy span //;
18  $Y(P(2n+1, 2n+1)) \leftarrow P(a(I_t)-0.5, b_{max}+0.5)$ ;
19  $\{P_1, P_2, \dots, P_m\} \leftarrow$  topologically sort all spans of  $\Phi(G_{s,t}) \cup P(2n+1, 2n+1)$ ;
20 Step 2: // pre-calculate all  $\Pr[F(i,k)]$  for  $1 \leq k \leq m$  and  $P_i \prec P_k$  //
21 for  $k \leftarrow 1$  to  $m$  step 1 do
22   for each  $P_j \in Y(P_k)$  do
23      $\Pr[F(j,k)] \leftarrow 1$ ; // the initial value of Eq. (4) //
24     for  $i \leftarrow 1$  to  $j$  step 1 with  $P_i \prec P_j$  do
25        $\Pr[F(i,k)] \leftarrow \Pr[F(i,j)]$ ; // the initial value of Eq. (5) //
26     end-for
27     for each  $I_v \in P_j \setminus P_k$  do
28        $\Pr[F(j,k)] \leftarrow \Pr[F(j,k)] \cdot q_v$ ; // Eq. (4) //
29       for  $i \leftarrow 1$  to  $j$  step 1 with  $P_i \prec P_j$  do
30         if  $(I_v \in P_i)$  then  $\Pr[F(i,k)] \leftarrow \Pr[F(i,k)] \cdot q_v$ ; // Eq. (5) //
31       end-for
32     end-for
33   end-for
34 end-for
35 Step 3: // compute  $\Pr[E(P_k)]$ ,  $1 \leq k \leq m$  //
36 for  $k \leftarrow 1$  to  $m$  step 1 do
37    $\Pr[E(P_k)] = 1 - \sum_{P_i \prec P_k} (\Pr[E(P_i)] \cdot \Pr[F(i,k)])$ ; //Eq. (3) //
38 end-for
39 return  $\Pr[E(P_m)]$ ;
end-algorithm

```

The following two trivial properties are required to prove Theorem 3.

Property 5. For spans $P(x,y)$, $P(x-1,y) \in \Phi(G_{s,t})$,

$$P(x-1,y) \setminus P(x,y) = \begin{cases} \{I_v\}, & \text{if there exists a probe } I_v \text{ such that } x-1 < b(I_v) < x, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Property 6. For spans $P(x,y)$, $P(x,y-1) \in \Phi(G_{s,t})$,

$$P(x,y-1) \setminus P(x,y) = \begin{cases} \{I_v\}, & \text{if there exists a non-probe } I_v \text{ such that } a(I_v) < x \text{ and } y-1 < b(I_v) < y, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Theorem 3. Given an n -vertex PIG G , the 2TR problem on G can be solved in $O(n^4)$ time.

Proof. Consider Algorithm 1. Notably, given a PIG G , the corresponding $D(G)$ can be constructed in linear time [8]. In Step 1, computing all spans $P(x,y)$ in lines 8 and 9 takes $O(n^3)$ time. Then computing all $Y(P(x,y))$ in lines 13 and 14 takes $O(n^2)$ time. Since the number of spans, $m=|\Phi(G_{s,t})|+1$, does not exceed $O(n^2)$ and checking whether $P' \prec P$, for any two spans P' and P , takes constant time, all spans can be sorted in topological order in $O(n^4)$ time in line 19.

In Step 2, let $P_k=P(x,y)$. By Property 4, P_j in line 22 is either $P(x-1,y)$ or $P(x,y-1)$. Therefore, the for-loop in line 22 is executed at most twice for each k . Thus, all initial values of $\Pr[F(j,k)]$ in line 23 and $\Pr[F(i,k)]$ in line 25 can be computed in $O(m)=O(n^2)$ time and $O(m \cdot m)=O(n^4)$ time, respectively. From Properties 5 and 6, the size of $(P_j \setminus P_k)$ in line 27 does not exceed one, so the body of the for-loop in line 27 is executed at most $O(m)$ times over the whole Step 2. Thus, all values of $\Pr[F(j,k)]$ in line 28 and $\Pr[F(i,k)]$ in line 30 can be updated in $O(m)=O(n^2)$ time and $O(m \cdot m)=O(n^4)$ time, respectively.

After all values of $\Pr[F(i,k)]$, $1 \leq k \leq m$ and $1 \leq i \leq k-1$, have been pre-calculated in Step 2, all values of $\Pr[E(P_k)]$ in line 37 are computed in $O(m^2)=O(n^4)$ time in Step 3. Therefore, the total time complexity of the algorithm is $O(n^4)$. $\square$

4. Conclusions

This paper proposes a polynomial-time algorithm for computing the 2-terminal reliability of probe interval graphs. However, determining the complexity of the K -terminal reliability of probe interval graphs remains an open problem.

References

- [1] H. M. AboElFotouh and C. J. Colbourn, Computing 2-terminal reliability for radio-broadcast networks. *IEEE Transactions on Reliability*, **38** No. 5 (1989), 538-555. <http://dx.doi.org/10.1109/24.46478>
- [2] H. M. AboElFotouh and C. J. Colbourn, Efficient algorithms for computing the reliability of permutation and interval graphs. *Networks*, **20** No. 7 (1990), 883-898. <http://dx.doi.org/10.1002/net.3230200706>
- [3] G. J. Chang, A. J. Klocks, J. Liu and S. L. Peng, The PIGs full monty—a floor show of minimal separators. In STACS (521-532), *Springer Berlin Heidelberg*, 2005. http://dx.doi.org/10.1007/978-3-540-31856-9_43
- [4] M. C. Golumbic, Algorithmic graph theory and perfect graphs. *Elsevier*, 2004. [http://dx.doi.org/10.1016/s0167-5060\(04\)80059-1](http://dx.doi.org/10.1016/s0167-5060(04)80059-1)
- [5] T. Klocks, C. S. Liu and S. L. Peng, Domination and Independent Domination on Probe Interval Graphs. *In the Proceedings of the 23rd Workshop on*

Combinatorial Mathematics and Computation Theory (2006), 93-97.

[6] M. S. Lin, A linear-time algorithm for computing K-terminal reliability on proper interval graphs. *IEEE Transactions on Reliability*, **51** No. 1 (2002), 58-62. <http://dx.doi.org/10.1109/24.994911>

[7] M. S. Lin and C. C. Ting, Computing K-terminal reliability of d-trapezoid graphs. *Information Processing Letters*, **113** No. 19-21 (2013), 734-738. <http://dx.doi.org/10.1016/j.ipl.2013.07.006>

[8] R. M. McConnell and Y. Nussbaum, Linear-time recognition of probe interval graphs. In Algorithms-ESA (349-360), *Springer Berlin Heidelberg*, 2009. http://dx.doi.org/10.1007/978-3-642-04128-0_32

[9] F. R. McMorris, C. Wang and P. Zhang, On probe interval graphs. *Discrete Applied Mathematics*, **88** No. 1 (1998), 315-324. [http://dx.doi.org/10.1016/s0166-218x\(98\)00077-8](http://dx.doi.org/10.1016/s0166-218x(98)00077-8)

[10] G. B. Mertzios, I. Sau and S. Zaks, A new intersection model and improved algorithms for tolerance graphs. *SIAM Journal on Discrete Mathematics*, **23** No.4 (2009), 1800-1813. <http://dx.doi.org/10.1137/09075994x>

[11] P. Zhang, Probe interval graphs and its applications to physical mapping of DNA. *Unpublished manuscript*, 1994.

Received: November 10, 2014; Published: January 7, 2015