

# Finding the $(k, l)$ –core of a tree network with unreliable edges

**Abdallah W. Aboutahoun**

Department of Mathematics, Faculty  
of Science, Alexandria University,  
Alexandria, Egypt.

**Eman Fares**

Department of Basic Sciences,  
Faculty of Engineering, Pharos  
University, Alexandria, Egypt.

## Abstract

Given a reliable tree network  $T$  containing  $n$  vertices where each edge has an independent operational probability, this paper presents finding a reliable subtree with at most  $k$  leaves and with a diameter of at most  $l$  which maximizes the expected number of nodes that are reachable from the selected subtree by operational paths. An efficient algorithm is presented for finding a reliable  $(k, l)$  – tree core of  $T$ . Numerical example is explained to clarify the efficient algorithm.

## 1. Introduction

The problem of finding a  $k$  –tree core is one of several types of location problems for a single facility on a tree which minimizes the sum of the distance. Regarding the problem of locating a tree shaped facility on trees, a generalization of the core problem is the  $k$  –tree core problem which involves finding a subtree that minimizes the sum of the distances and which contains exactly  $k$  leaves. [1] presented an algorithm for constructing a  $k$  –tree core on trees which has time complexity of  $O(kn)$ , where  $n$  is the number of vertices of the tree. [2] designated a linear time algorithm for finding a  $k$  –tree core with a time complexity  $O(n)$ . [3] developed a time algorithms with a time complexity of  $O(n^2 \log n)$  for weighted trees and with a time complexity of  $O(n^2)$  for unweighted trees. The main idea of these two algorithms is that, by giving the tree  $T$ , the first new rooted trees  $T_{r_i}$  are constructed where the maximum length of a path is at most  $l$ .

Then, for each subtree  $T_{r_i}$ , they proposed a subtree containing the root with at most  $k$  leaves. It minimizes the sum of the distances. [4] introduced two algorithms for finding a  $(k, l)$  subtree core of  $T$ . The first algorithm had a time complexity of  $O(n^2)$  for all edges of any length. The second algorithm had a time complexity of  $O(lkn)$  for the edges of 1 length. [5] provided an algorithm that show the 2 unreliable path location problem on tree networks in  $O(n^2)$  time. [6] presented efficient algorithms for finding a core of a tree with a specified length. The first algorithm had a time complexity of  $O(n \log n)$  and the second is  $O(\log^2 n)$  using  $O(n)$  processors on an EREW PRAM model.

This paper focuses on the location of a reliable  $(k, l)$  – tree core with unreliable edges with a length of at most  $l$  and having  $k$  leaves that maximize the reachability of the demand nodes to this location.

The remainder of this paper is organized into four sections. In Section 2, we give notation and formulation. In Section 3, we present an efficient algorithm for finding a reliable  $(k, l)$  – tree core. In Section 4, we present example is provided to illustrate our algorithm. Finally, in Section 5, we give concluding remarks.

## 2. Notation and Formulation

Let  $T = (V, E)$  be unweighted tree, where  $V$  is the vertex set,  $|V| = n$  and  $E$  is the edge set,  $|E| = n - 1$ . Let  $d(v, u)$  be an unweighted length between each edge  $(u, v) \in E$ . Let  $P$  be a path in tree  $T$ , where  $v$  and  $u$  are two vertices in. The unique path from  $v$  to  $u$  is considered as the form  $P(v, u)$ , where each edge has an independent operational probability. The probability of success of each edge is denoted as the form  $p_e(v, u)$  while the probability of fail of each edge is denoted as the form  $q_e(v, u)$  such that  $1 - p_e(v, u) = q_e$ . all vertices are assumed to be perfectly reliable and any vertex is called a leaf vertex if the number of edges incident to it is equal to

**1.** The diameter of  $T$  is  $\text{Max}_{u, v \in V} d(v, u)$  while the height of is  $\text{Max}_{u \in V} d(r, u)$ , where  $r$  is the root of tree  $T$ . A reliable  $(k, l)$  is a subtree with at most  $k$  leaves and with a diameter of at most  $l$  which maximizes the expected number of nodes that are reachable from the selected subtree by operational paths.

**Definition 1** Under the assumption that edges fail independently, the probability that the path  $P(v, w)$  is operational at a random point in time is

$$R(v, w) = \prod_{e \in P(v, w)} p_e \quad (1)$$

and the probability that at least one edge in the path has failed deeming the path unusable is

$$1 - R(v, w) = 1 - \prod_{e \in P(v, w)} p_e \quad (2)$$

Suppose that the location of the point  $k$  can be at any node or edge of the network. If  $k$  is located on an edge  $e = (v, w)$  then splitting the edge  $e$  into two new edges: edge  $(v, k)$  with length  $x$  and edge  $(k, w)$  with length  $d_e - x$ , with operational probability  $p(v, w)$  and  $p(k, w)$ , respectively. Let  $X(k, w)$  be the Bernoulli random variable that denotes whether or not there is an operational path from  $k$  to  $w$  at a random instance, where

$$Pr(X(k, w) = 1) = Pr(P(k, w)) = R(k, w)$$

**Definition 2** We define the reliable 1 –median problem (Relisum) as the form

$$\min_{w \in V} \sum_{w \in V} R(k, w) \quad (3)$$

**Definition 3** We define the reliable sum  $R(k, P)$  from any vertex  $k$  which does not belong to the path  $P$  to the path  $P$  on tree  $T$  such that

$$S(P) = \sum_{k \notin P} R(k, P) \quad (4)$$

where  $R(k, P)$  is the maximum reliability from node  $k \notin P$  to a vertex  $i$  on  $P$

$$R(k, P) = \prod_{e \in P(k, w)} p_e \quad (5)$$

**Definition 4** For a rooted tree  $T$  the total connectivity of any vertex  $v_i \in V$  is defined as the form

$$R(v_i) = \sum_{u_j \in V} P(v_i, u_j) = 1 + \sum_{u_j \in V \setminus \{v_i\}} R(v_i, u_j) \quad (6)$$

where  $P(v_i, v_i) = 1$

**Definition 5** For a rooted tree  $T$  the total connectivity of any path  $p$  is the sums of the reliability from all other vertices in the network to the path such that  $v \notin p$ .

$$R(P) = \sum_{v \notin P} R(v, p) \quad (7)$$

where  $R(v, p)$  is the maximum reliability from  $v \notin p$  to a vertex of  $P$

**Theorem 1** For each vertex  $v \in V$ , we define recursively the reliability sum of the vertices in the subtree  $T_\alpha^v$  to  $v$  or the expected number of nodes in  $V_\alpha(v)$  which can be reached from node  $v$  as follows

$$R_\alpha(v) = 1 + \sum_{v_k \in C(v)} R_\alpha(v_k) p(v, v_k) \quad (8)$$

and the reliability sum of the vertices in  $T_\beta^v$  to  $v$  or the expected number of nodes in  $V_\beta(v)$  which can be reached from node  $v$  as follows

$$R_\beta(v) = \left[ R_\beta(f(v)) + 1 + \sum_{\substack{v_t \in C(f(v)) \\ v_t \neq v}} R_\alpha(v_t) p(v_k, f(v)) \right] p(f(v), v) \quad (9)$$

where  $C(v)$  denote the set of nodes which consists of all children of  $v$ ,  $R_\alpha(v) = 1$  if  $v$  is a leaf and  $R_\beta(v) = 0$  if  $v$  is the root of the tree. So,  $R_\alpha(v)$  and  $R_\beta(v)$  denote the

expected number of nodes in  $V_\alpha(v)$  and  $V_\beta(v)$  respectively which can be reached from node. The labels  $R_\alpha(v)$  and  $R_\beta(v)$  can be evaluated recursively at each vertex of  $T^r$ . In particular, in a bottom up visit of nodes of the tree we first associate with each vertex  $v$  the label  $R_\alpha(v)$  by beginning with  $R_\alpha(v) = 1$  for all leaves  $v$  on the other hand, by proceeding up down from the root to the leaves,  $R_\beta(v)$  can be evaluated, beginning with  $R_\beta(r) = 0$ .

**Lemma 1** For each vertex  $v_i \in V$ , we define the total connectivity sum for any vertex  $v_i$  as the form (10)

$$R(v_i) = R_\alpha(v_i) + R_\beta(v_i)$$

**Lemma 2** For all edge  $e = (u_i, v_i)$ , where  $u_i = \text{tail}(e)$  and  $v_i = \text{head}(e)$ , the total reliable sum of edge  $e$  define as the form

$$R(e = (u_i, v_i)) = (1 - p_e(u_i, v_i))R_\alpha(v_i) + R(u_i) \quad (11)$$

**Proof**

$$\begin{aligned} R_e(u_i, v_i) &= R_{\text{head}(e)}(v_i) + R_{\text{tail}(e)}(u_i) \\ &= R_\alpha(v_i) + R(u_i) - p_e(u_i, v_i)R_\alpha(v_i) \\ &= (1 - p_e(u_i, v_i))R_\alpha(v_i) + R(u_i) \end{aligned}$$

where  $R_{\text{head}(e)}(v_i) = R_\alpha(v_i)$  and  $R_{\text{tail}(e)}(u_i) = R(u_i) - p_e(u_i, v_i)R_\alpha(v_i)$

The relisum of the path- shaped facility with only one edge  $R(e)$  can be evaluated also by

$$\begin{aligned} R(e = (u_i, v_i)) &= R(u_i: p(e) = 1) = R(v_i: p(e) = 1) \\ &= 2 + \sum_{v_k \in C(v_i)} R_\alpha(v_k) p_e(v_i, v_k) \\ &\quad + \sum_{\substack{w_t \in C(u_i) \\ w_t \neq v_i}} R_\alpha(w_t) p((u_i, w_t)) + R_\beta(u_i) \end{aligned}$$

**Lemma 3** The reliability of the network which we will call relisum when the path  $P(v, u)$  is located where the node  $u$  is adjacent to  $v$  is as follows

$$\begin{aligned} R(P(v, u)) &= 2 + \sum_{v_k \in C(v) \setminus u} R(v_k) p(v, v_k) \\ &\quad + \sum_{v_j \in C(u)} R(v_j) p(u, v_j) \\ &= \sum_{u \in V} \prod_{(v_i, v_j) \in P(u, v)} p(v_i, v_j) \end{aligned} \quad (12)$$

where  $p(v, v) = 1$

**Lemma 4** For all  $x \in V_\alpha(v)$ , the relisum of the path  $P_{vx}$  in  $T_\alpha^v$  is

$$R_\alpha(P_{vx}) = R_\alpha(v) + RSAV_\alpha(v, P_{vx}) \quad (13)$$

**Proof**

Let  $P_{vx}$  be a path connecting the root  $v$  of the rooted tree  $T_\alpha^v$  to any vertex  $x \in V_\alpha(v)$ . Suppose that this path is consisting of the sequence of vertices  $\{v = v_1, v_2, \dots, v_h = x\}$ . As we defined above

$$\begin{aligned} RSAV_\alpha(v, P_{vx}) &= (1 - p(v, v_2)) R(v_2) \\ &+ (1 - p(v_2, v_3)) R(v_3) + \dots \\ &+ (1 - p(v_{h-2}, v_{h-1})) R(v_{h-1}) + (1 - p(v_{h-1}, x)) R(x) \end{aligned}$$

The reliable sum  $R_\alpha(v_i)$  of each vertex  $v_i$  on the path  $P_{vx}$  can be evaluated using the following formula

$$R_\alpha(v_i) = 1 + \sum_{v_k \in \mathcal{C}(v)} R_\alpha(v_k) p(v_i, v_k)$$

The reliable sum  $R_\alpha(P_{vx})$  of the location of the path  $P_{vx}$  in the subtree  $T_\alpha^v$  is the sum of  $R_\alpha(v_i)$  of the vertices of the path  $P_{vx}$  excluding the edges of the path.

$$\begin{aligned} R_\alpha(P_{vx}) &= 1 + \\ &\sum_{v_{k1} \in \mathcal{C}(v_1) \setminus \{v_2\}} R_\alpha(v_{k1}) p(v_1, v_{k1}) + 1 + \\ &\sum_{v_{k2} \in \mathcal{C}(v_2) \setminus \{v_3\}} R_\alpha(v_{k2}) p(v_2, v_{k2}) + \dots + 1 \\ &+ \sum_{v_{kh-1} \in \mathcal{C}(v_{h-1}) \setminus \{v_h\}} R_\alpha(v_{kh-1}) p(v_{h-1}, v_{kh-1}) + 1 \\ &+ \sum_{v_{kh} \in \mathcal{C}(v_h)} R_\alpha(v_{kh}) p(v_h, v_{kh}) \end{aligned}$$

This can be written in the form

$$\begin{aligned} R_\alpha(P_{vx}) &= 1 + \\ &\sum_{v_{k1} \in \mathcal{C}(v_1)} R_\alpha(v_{k1}) p(v_1, v_{k1}) - R_\alpha(v_2) p(v_1, v_2) + 1 \\ &+ \sum_{v_{k2} \in \mathcal{C}(v_2)} R_\alpha(v_{k2}) p(v_2, v_{k2}) - R_\alpha(v_3) p(v_2, v_3) \\ &+ \dots + 1 + \sum_{v_{kh-1} \in \mathcal{C}(v_{h-1})} R_\alpha(v_{kh-1}) p(v_{h-1}, v_{kh-1}) \\ &- R_\alpha(v_h) p(v_{h-1}, v_h) + 1 \\ &+ \sum_{v_{kh} \in \mathcal{C}(v_h)} R_\alpha(v_{kh}) p(v_h, v_{kh}) \end{aligned}$$

Hence,

$$\begin{aligned} R_\alpha(P_{vu}) &= R_\alpha(v_1) - R_\alpha(v_2) p(v_1, v_2) + R_\alpha(v_2) \\ &- R_\alpha(v_3) p(v_2, v_3) + \dots + R_\alpha(v_{h-1}) \\ &- R_\alpha(v_h) p(v_{h-1}, v_h) + R_\alpha(v_h) \\ &= R_\alpha(v_1) + (1 - p(v_1, v_2)) R_\alpha(v_2) \\ &+ \dots + (1 - p(v_{h-1}, v_h)) R_\alpha(v_h) \\ &= R_\alpha(v) + SAV_\alpha(v, P_{vx}) \end{aligned}$$

**Lemma 5** For any  $v, u \in V$  the reliable saving of  $P(v, u)$ , as the form

$$RSAV_\alpha(P(v, u)) = R_\alpha(P(v, u)) - R_\alpha(v) \quad (14)$$

Recursively

$$\begin{aligned} RSAV_\alpha(P(v, u)) &= \\ \begin{cases} SR(P(v, u)) = (1 - p(v, u)) R_\alpha(u), & v = f(u) \\ RSAV_\alpha(P(v, f(u))) + SR(P(u, f(u))), & v \neq f(u) \end{cases} \end{aligned} \quad (15)$$

**Definition 6** Let  $P(v, u)$  be a path in  $T_r$  in the form  $\{v = v_1, v_2, \dots, v_h = u\}$  then we have,

$$\begin{aligned} R_\alpha(P(v, u)) &= \\ &\sum_{i=1}^{n-1} [R_\alpha(v_i) - R_\alpha(v_i + 1) p(v_i, v_{i+1})] + R_\alpha(v_n) \end{aligned} \quad (16)$$

and

$$\begin{aligned} RSAV_\alpha(v, P(v, u)) &= \\ &= \sum_{i=1}^{n-1} [1 - p(v_i + 1) p(v_i, v_{i+1})] R_\alpha(v_i + 1) \end{aligned} \quad (17)$$

**Lemma 6** Let  $P_{vu}$  be a path in a tree  $T_v$ . The reliability saving of the path  $P_{vu}$  to be as the form

$$RSAV_\alpha(v, P_{vu}) = RSAV_\alpha(v, P_{vf(u)}) + R_{Edge}(P_{f(u)u}) \quad (18)$$

where  $R_{Edge}(P_{f(u)u}) = (1 - P_{f(u)u}) R_\alpha(u)$

**Proof**

$$\begin{aligned} RSAV_\alpha(v, P_{vu}) - RSAV_\alpha(v, P_{vf(u)}) &= \\ &= (R_\alpha(P_{vu}) - R_\alpha(v)) \\ &- (R_\alpha(v, P_{vf(u)}) - R_\alpha(v)) \\ &= R_\alpha(P_{vu}) - R_\alpha(P_{vf(u)}) \\ &= (1 - P_{f(u)u}) R_\alpha(u) \end{aligned}$$

**Lemma 7** Let  $P(u, v)$  be any path in a tree  $T$  and let  $S$  be any subtree of  $T$  which connected with a path  $P(u, v)$  at node  $u$  then the relisum of  $(S^* = S \cup P(v, u))$  is as the form

$$R(S^*) = R(T) + RSAV_\alpha(v, P(v, u)) \quad (19)$$

**Proof**

$$\begin{aligned} R(S^*) - R(S) &= \sum_{i \in V} R(i, S \cup P(v, u)) - \sum_{i \in V} R(i, S) \\ &= \sum_{i \in V(T_v(v_2))} (R(i, P(v, u)) - R(i, v)) + \\ &\sum_{i \notin V(T_v(v_2))} (R(i, S) - R(i, S)) \end{aligned}$$

**Lemma 8** Let  $P(v, u)$  be a path from vertex  $v$  to vertex  $u$  in a tree  $T$  and  $S$  is any subtree of  $T$  such that  $(v, u) \cap S = \{v\}$  or  $P(v, u) \cap S = \{u\}$ , then

$$R(S \cup P(v, u)) - R(S) = R_\alpha(P(v, u)) + R_\alpha(v) \quad (20)$$

**Definition 7** [3] Given a path  $P$  having an even or odd length  $l$ , the midpoint of  $P$  is a vertex or edge whose removal divides  $P$  into two paths with length  $l/2(l - 1/2)$ .

### 3. Algorithm

In this section, an algorithm for finding the location of a reliable  $(k, l)$  –tree core facility in a tree network with unreliable edges is presented. The algorithm uses the technique bottom-up fashion. Orient the input tree  $T$  into a rooted tree  $T^c$ .  $c$  is the root of  $T^*$  and has degree at least 2.  $T^*$  has exactly  $k$  leaves. The reliability sum of the subtree  $T^*$   $R(T^*)$  is maximized.

We give an example to demonstrate how the algorithm works, From figure (1) we take all paths  $P$  which has some lengths is  $\leq l$  and  $k = 4$ .

For  $l = 6$  (even length)

let  $P = ((2,5), (1,2), (1,4), (4,10), (10,18), (18,28))$ , Orient the tree form center of  $P$  ( node 4 ), from equation (16) we calculate  $P_1 = (4,1), (1,3), (3,7)$  and  $P_2 = (4,11), (11,20), (20,30)$ , where  $P_1$  is the first maximum saving and  $P_2$  is the second maximum saving. Then we find  $(P_1 \cup P_2)$  Core =  $((3,7), (1,3), (4,1), (4,11), (11,20), (20,30))$ , then we construct induced subtree  $T^{*4} = \langle 6,7,8,30 \rangle$ . Finally from equation (19) we calculate the reliability sum of induced subtree  $R(T^*) = 22.7734$

For  $l = 7$  (odd length)

let  $P = (5,12), (2,5), (1,2), (1,4), (4,10), (10,18), (18,21)$ , Orient the tree form center of  $P$  is edge  $(1,4)$ ,  $P_1 = ((1,4), 2), (2,6), (6,14)$ ,  $P_2 = ((1,4), 3), (3,7), (7,15)$  where  $P_1$  is the first maximum saving and  $P_2$  is the second maximum saving, then we find  $(P_1 \cup P_2)$  Core =  $((6,14), (2,6), (1,2), (1,4), (4,3), (3,7), (7,15))$ , then we construct induced subtree  $T^{*(1,4)} = \langle 14,15,16,29 \rangle$ . Finally from equation (19) we calculate the reliability sum of induced subtree  $R(T^*) = 19.1643$ .

#### 3.1 Algorithm for finding $(k, l)$ – subtree core $T^*$ of the tree $T$

**INPUT:** An unweighted tree  $T$

**OUTPUT:** A  $(k, l)$  – subtree core  $T^*$  which maximizes the reliability sum  $R(T^*)$  of the tree  $T$  begin

**Step1:** find all paths with length  $\leq l$  and  $k = 2$

**Step2:** if  $l$  is (even length) then orient the input tree  $T$  into a rooted tree  $T^c$  where  $c$  is the center of  $P$  else orient the input tree  $T$  into a rooted tree  $T^e$  where  $e$  is the center edge of  $P$

**Step3:** for  $T^c$  find  $P_1$  and  $P_2$  with length  $\leq l/2$  and for  $T^e$  find  $P_1$  and  $P_2$  with length  $\leq (l - 1)/2$ , s.t Core =  $(P_1 \cup P_2)$

**Step4:** for  $C = 2:l$  and for each  $v_i \in V(P)$  compute  $Q(v_i) = \{Reliable\ saving\ at\ vertex\ v_i\}$   
s.t  $(C \cup Q(v_i)) \leq l$

**Step5:** for all  $v_i$  find  $\max_v Q(v_i)$   
s.t  $V(C) \cup V(Q(v_i))$

**Step6:** find all Induced Subtrees  $T^*$  of the tree  $T$  and compute reliable of all subtrees  $R(T^*)$ .

### 4. Numerical Example

In this section, an example of the proposed algorithm is discussed. A network tree with  $n = 40$  vertices and  $E = 39$  edges is considered in Figure (1), each edge has an arbitrary positive probabilities. For  $l$  is any Even or Odd length and  $k = 4$ .

We begin by evaluating for each vertex  $v \in V$  the two quantities  $R_\alpha(v)$  and  $R_\beta(v)$  that represent the expected number of nodes in  $V_\alpha(v)$  which can be reached from node  $v$ , and the expected number of nodes in  $V_\beta(v)$  which can be reached from node  $v$ . So, table 1 gives the values of the three quantities  $R_\alpha(v)$ ,  $R_\beta(v)$  and  $R(v) = R_\alpha(v) + R_\beta(v)$  for all vertices  $v \in V$  These values are used in calculating all other quantities in the algorithm. When a single edge path-shaped facility  $e = (u, v)$  to be located, three quantities  $R_{head}(e) = R_\alpha(v)$ ,  $R_{tail}(e) = R(u) - p(e)R_\alpha(v)$  and  $R(e) = R_{head}(e) + R_{tail}(e)$  are calculated in table 2, in table 3 gives  $P_1$  and  $P_2$ , where  $P_1$  is the first path with MAXSAV and  $P_2$  is the second path with MAXSAV, table 4 presents: The induced subtrees with maximum reliability saving and the reliability sum of each subtree.

### 5. Conclusion

In this paper, we presented our algorithms for finding a reliable  $(k, l)$  –core of tree networks with unreliable edge. The algorithm can be readily extended to a tree network in which edges have length equal 1. So, given two parameters  $k$  and  $l$ , the objective is to locate a reliable  $(k, l)$  –tree core which is a subtree with a diameter  $l$  and having  $k$  leaves such that the sum of the reliabilities from all the vertices of the tree network to this facility is maximized. The objective of the problem is to maximize the expected number of nodes reachable by operational paths from the located site.

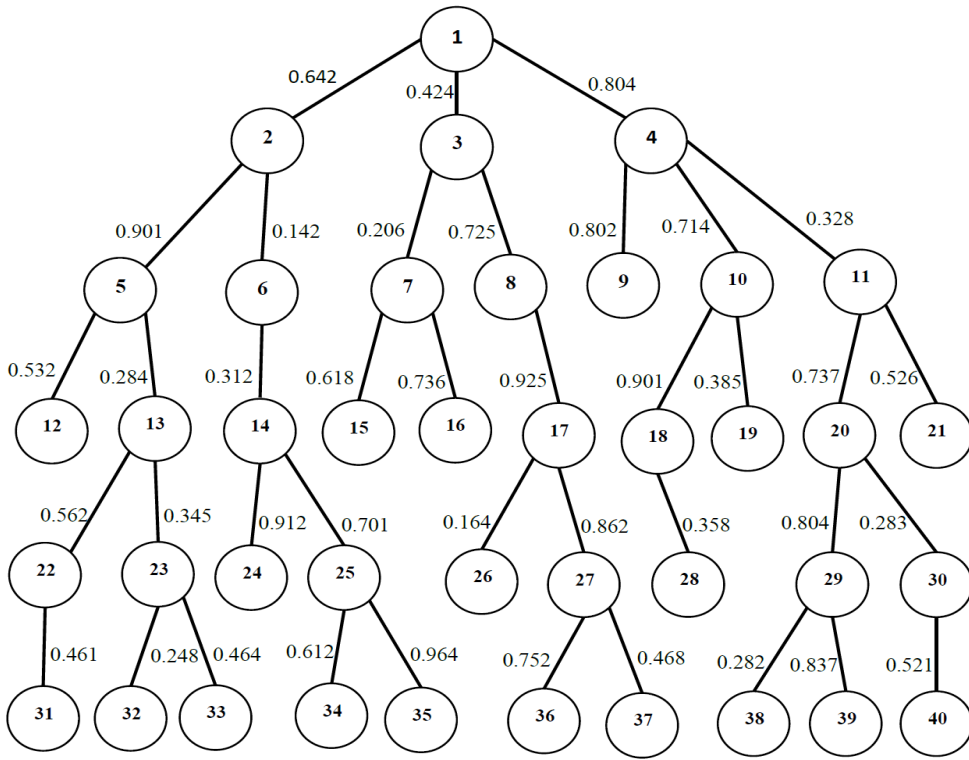


Figure 1: A tree example on which the proposed algorithm is applied where the edge labels are the operational probabilities.

Table 1: The values of  $R_\alpha(v)$ ,  $R_\beta(v)$  and  $R(v)$  for all nodes  $v \in V$

$v$	1	2	3	4	5	6	7	8	9	10
$R_\alpha(v)$	8.8912	3.3042	4.2739	4.9227	2.2169	2.1599	2.3540	3.8468	1	2.6086
$R_\beta(v)$	0	4.3463	3.0015	3.9664	5.0934	1.0428	1.3988	3.2527	6.4859	5.0170
$R(v)$	8.8912	7.6505	7.2754	8.8891	7.3103	3.2028	3.7528	7.0995	7.4859	7.6256
$v$	11	12	13	14	15	16	17	18	19	20
$R_\alpha(v)$	3.8358	1	2.4117	3.7178	1	1	3.0776	1.3580	1	3.1341
$R_\beta(v)$	2.5030	3.6061	1.8816	0.6374	1.9373	2.2204	3.9337	5.7682	2.7876	2.9693
$R(v)$	6.3388	4.6061	4.2933	4.3551	2.9373	3.2204	7.0114	7.1262	3.7876	6.1035
$v$	21	22	23	24	25	26	27	28	29	30
$R_\alpha(v)$	1	1.4610	1.7120	1	2.5760	1	2.2200	1	2.1190	1.5210
$R_\beta(v)$	3.0575	1.9514	1.2774	3.1401	1.7871	1.1230	4.3942	2.4230	3.5374	1.6055
$R(v)$	4.0575	3.4124	2.9894	4.1401	4.3631	2.1230	6.6142	3.4230	5.6564	3.1265
$v$	31	32	33	34	35	36	37	38	39	40
$R_\alpha(v)$	1	1	1	1	1	1	1	1	1	1
$R_\beta(v)$	1.3606	0.6799	1.1718	2.2957	3.2767	4.4084	2.8764	1.5156	4.0339	1.3574
$R(v)$	2.3606	1.6799	2.1718	3.2957	4.2767	5.4084	3.8764	2.5156	5.0339	2.3574

Table 2: The reli-sums of paths of length 1 calculated by summing  $R_{tail}(e)$  and  $R_{head}(e)$

Edge $e$	(1, 2)	(1, 3)	(1, 4)	(2, 5)	(2, 6)	(3, 7)	(3, 8)	(4, 9)
$R_{head}(e)$	3.3042	4.2739	4.9227	2.2169	2.1599	2.354	3.8468	1
$R_{tail}(e)$	6.7699	7.0791	4.9334	5.3653	7.3438	6.7905	4.4865	8.0871
$R(e)$	10.0741	11.353	9.8561	7.8699	9.5037	9.1445	8.3333	9.0871
Edge $e$	(4, 10)	(4, 11)	(5, 12)	(5, 13)	(6, 14)	(7, 15)	(7, 16)	(8, 17)
$R_{head}(e)$	2.6086	3.8358	1	2.4117	3.7178	1	1	3.0776
$R_{tail}(e)$	7.0266	7.631	6.7783	6.6254	2.0428	3.1348	3.0168	4.2527
$R(e)$	9.6352	11.4668	7.7783	9.0371	5.7606	4.1348	4.0168	7.3303
Edge $e$	(10,18)	(10,19)	(11,20)	(11,21)	(13,22)	(13,23)	(14,24)	(14,25)
$R_{head}(e)$	1.358	1	3.1341	1	1.461	1.712	1	2.576
$R_{tail}(e)$	6.402	7.2406	4.029	5.8128	3.4722	3.7027	3.4431	2.5494
$R(e)$	7.76	8.2406	7.1631	6.8128	4.9332	5.4147	4.4431	5.1254
Edge $e$	(17,26)	(17,27)	(18,28)	(20,29)	(20,30)	(22,31)	(23,32)	(23,33)
$R_{head}(e)$	1	2.22	1	2.119	1.521	1	1	1
$R_{tail}(e)$	6.8474	5.0977	6.7686	4.3998	5.673	2.9514	2.7414	2.5254
$R(e)$	7.8474	7.3177	7.7682	6.5188	7.194	3.9514	3.7414	3.5254
Edge $e$	(25,34)	(25,35)	(27,36)	(27,37)	(29,38)	(29,39)	(30,40)	-
$R_{head}(e)$	1	1	1	1	1	1	1	-
$R_{tail}(e)$	3.7511	3.3991	5.822	6.1462	5.3744	4.8194	2.6055	-
$R(e)$	4.7511	4.3991	6.8621	7.1462	6.3744	5.8194	3.6055	-

Table 3: Some paths to find  $P_1$  and  $P_2$

Path	Even or Odd	Center of path	$P_1$ =First path with <i>MAXSAV</i>	$P_2$ =Second path with <i>MAXSAV</i>
(6,14), (2,6), (1,2), (1,4), (4,11), (11,20)	Even	1	(1,2), (2,6), (6,14)	(1,3), (3,7), (7,15)
(2,5), (1,2), (1,4), (4,10), (10,18), (18,28)	Even	4	(4,1), (1,3), (3,7)	(4,11), (11,20), (20,30)
(5,12), (2,5), (2,6), (6,14), (14,25), (25,34)	Even	6	(6,2), (2,1), (1,3)	(6,14), (14,25), (25,34)
(25,35), (14,25), (6,14), (2,6), (1,2), (1,4), (4,10)	Odd	(2,6)	((2,6),1), (1,3), (3,7)	((2,6),14), (14,25), (25,34)
(5,12), (2,5), (1,2), (1,4), (4,10), (10,18), (18,21)	Odd	(1,4)	((1,4),2), (2,6), (6,14)	((1,4),3), (3,7), (7,15)
(17,26), (8,17), (3,8), (1,3), (1,4), (4,11), (11,21)	Odd	(1,3)	((1,3),2), (2,6), (6,14)	((1,3),4), (4,11), (11,20)

Table 4: The induced subtrees with maximum reliability saving and the reliability sum of each subtree

$Core = (P_1 \cup P_2)$	Induced Subtree $T^*$	$R(T^*)$
(6,14), (2,6), (1,2), (1,3), (3,7), (7,15)	<14,15,17,20>	24.8535
(3,7), (1,3), (4,1), (4,11), (11,20), (20,30)	<6,7,8,30>	22.7734
(25,34), (14,25), (6,14), (6,2), (2,1), (1,3)	<13,3,4,34>	20.9972
(25,34), (14,25), (6,14), (2,6), (2,1), (1,3), (3,7)	<34,7,8,11>	23.5746
(6,14), (2,6), (1,2), (1,4), (4,3), (3,7), (7,15)	<14,15,16,29>	19.1643
(6,14), (2,6), (1,2), (1,4), (4,11), (11,20)	<13,14,20,21>	22.7339

## References

- [1] S. Peng, A.B. Stephens, Y. Yesha, "Algorithms for a core and a  $k$ -tree core of a tree", J. Algorithms Vol. 15, pp. 143-159, (1993).
- [2] A. Shioura and T. Uno, "A linear time algorithm for finding a  $k$ -tree core", J Algorithms, Vol. 23 pp. 281- 290, (1997).
- [3] R.I. Becker, I. Lari, G. Storchi, A. Scozzari, "Efficient algorithms for finding the  $(k, l)$ -core of tree networks", Networks Vol. 40(4), pp. 208-251, (2002).
- [4] B. Wang, S.Peng, H.Yu and S.Ku, "Efficient algorithms for a constrained  $k$ -tree core problem in a tree network", Journal of Algorithms, Vol. 59, pp. 107-124, (2006).
- [5] J. Puerto, F. Ricca and A. Scozzari, "Reliability problems in multiple path-shaped facility location on networks", Discrete Optimization, Vol. 12, pp. 61-72, (2014).
- [6] S. Peng, W. Lo, "Efficient algorithms for finding a core of a tree with a specified length", J. Algorithms, Vol. 20, pp. 445-458, (1996).