RECEIVED
NOV 02 1998
OSTI

*Maximum-Path Leaves Relative to Vertices
and the Vertex One Center of a Spanning Tree:
An Enumeration and Analysis
of Configurations*

Los Alamos
NATIONAL LABORATORY

*Los Alamos National Laboratory is operated by the University of California
for the United States Department of Energy under contract W-7405-ENG-36.*

*This work was supported by the U.S. Department of Energy
and by the U.S. Department of Defense, Joint Program Office for
Biological Defense*

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither The Regents of the University of California, the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of The Regents of the University of California, the United States Government, or any agency thereof. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

*Maximum-Path Leaves Relative to Vertices
and the Vertex One Center of a Spanning Tree:
An Enumeration and Analysis
of Configurations*

L. Jonathan Dowell

Maximum-Path Leaves Relative to Vertices and the Vertex One Center of a Spanning Tree: An Enumeration and Analysis of Configurations

by

L. Jonathan Dowell

Abstract

This paper enumerates the configurations of a pair of vertices (one of which is a leaf) relative to the unique vertex one center and the leaf of greatest path length from this center in a spanning tree of a graph. From this enumeration, it demonstrates that the vertex one center is between the vertex and the leaf of greatest path length from the vertex for every such pair of vertices. Using this result, this paper develops an algorithm for the optimal reconfiguration of a tree infrastructure required to recover from the failure of a network element.

Introduction

The centers of a graph are an important concept in graph theory and its applications [13, 16, 17]. The identification of centers is often used to optimize the location of a facility in a network [3–5, 9]. The absolute one center of a graph can be used to discover a spanning tree of minimum diameter [6]. This relationship between a spanning tree and its centers is important for spanning-tree applications in infrastructure optimization, such as those found in communications and computer networks [14, 15].

This paper examines the configurations of vertices and leaves relative to the vertex one center (V1C) of a spanning-tree graph and the vertex having the greatest path length from this V1C. The paper enumerates the possible combinations of positions of vertices and leaves relative to these vertex one center characteristics and examines the allowed configurations of vertices and the leaves in the spanning tree having the greatest path lengths from the vertices. The result uses a novel method to confirm an important finding by Handler [7]. Finally, the results are applied to the problem of optimal reconfiguration of an infrastructure following the disruption of the infrastructure by failures of network elements.

Analyses of the Vertices of a Spanning Tree

Consider a spanning tree T of a graph $G(V, E)$ with vertices V and edges E . Associated with each edge $e \in E$ is a positive-valued function $l(e)$ called the length of e . The length of a path $a_1 a_2$ between vertices a_1 and a_2 is the sum of the lengths of the edges

belonging to the unique path P between a_1 and a_2 in T :

$$a_1 a_2 = \text{length of path } P = \sum_{e \in P} l(e). \quad (1)$$

Every vertex v in T partitions T into λ parts, where λ is the degree of v . Each edge adjacent to v starts a branch of T rooted at v . The vertex v is the only vertex of attachment between these branches. **Proof:** Consider vertices a_1 in one branch and a_2 in another branch. There exists a path from a_1 to v through the branch containing a_1 . Similarly, there exists a path from a_2 to v through the branch containing a_2 . Therefore, there is a path in T from a_1 to a_2 that contains v . This path is unique, since T is a tree. Consequently, there are no other paths in T between the vertices of the two branches rooted at v , other than through v . Ergo, there are no other vertices of attachment between the branches rooted at v , other than the vertex v .

The spanning tree has a VIC at vertex b . For each vertex v in T , another vertex that is farthest from it defines a maximum path length from v . The VIC is the vertex b such that the maximum path length from b is the minimum of all vertices in T .

Theorem 1: T can have at most two VICs.

Proof: Figure 1 illustrates T for this proof. The VIC b partitions T into λ parts, where λ is the degree of b . The leaf c is the vertex farthest from b . Call the branch containing c the maximal branch of T . Call all other branches submaximal branches. First, let $\alpha \neq b$ be any vertex of a submaximal branch of T . Then the path length αc is greater than the path length bc , because the path length αb is strictly positive. Therefore, since $\alpha c > bc$, α cannot be a VIC. Next, let β be the vertex in the path between b and c that is closest to b . There exists a vertex d in a submaximal branch of T such that $\beta d \geq bc$. Because β is between b and c , $\beta c < bc$. Since β is between b and all vertices of the maximal branch, the path length from β to any vertex in the maximal branch is less than the path length from b to that vertex. But since b is a VIC, there can be no vertex in T with a maximum path length less than bc . Therefore, there must be a vertex d in a submaximal branch with path length $\beta d \geq bc$. The vertex β is a VIC if and only if $\beta d = bc$ and d is the vertex in T that is the farthest from β . For each other vertex $\gamma \neq b$ in the maximal branch, $\gamma d = \gamma\beta + \beta d > \beta d$, therefore γ cannot be a VIC. Since there can be only one vertex β that is closest to b in the maximal branch, there can be at most two VICs in T . QED.

Figure 1. A spanning tree T with VIC b .

For further discussion, let b be a VIC of T . The leaf c is the vertex farthest from b , defining the maximal branch of T . That is, $bc \geq bv$ for each vertex v in T .

Figure 2 shows an enumeration of the possible locations of a vertex a relative to the VIC b and the maximal-branch leaf c that is farthest from b in T . Either a is in a submaximal branch (a_1), or a is in the maximal branch which contains c . If a is in the maximal branch, then either a is on the path between b and c (a_2) or a is on a branch off this path (a_3).

Figure 2. A spanning tree T with VIC b .

Next, consider the possible locations for a vertex b' and a leaf d relative to b and c . The vertex b' can occupy positions of types a_1 , a_2 , and a_3 . Because d is a leaf, it cannot be a vertex on the path between b and c , unless $d = c$. Table 1 lists explicitly the possible combinations of position types for vertices b' and d . Using this enumeration, there are 19 possible configurations of the vertices b' and d relative to b and c . Table 2 lists explicitly the possible configurations of the vertices b' and d in T .

Table 1. Combinations of position types for a vertex b' and leaf d relative to b and c in T .

Vertex b'	Leaf d
a_1	a_1
a_1	a_2
a_1	a_3
a_2	a_1
a_2	a_3
a_2	a_2
a_3	a_1
a_3	a_2
a_3	a_3

Table 2. Enumeration of position types for a vertex b' and leaf d relative to b and c in T .

Position-type combination $b' = a_1$ and $d = a_1$

Position-type combination $b' = a_1$ and $d = a_2$

Position-type combination $b' = a_1$ and $d = a_3$

Position-type combination $b' = a_2$ and $d = a_1$

Position-type combination $b' = a_2$ and $d = a_2$

Table 2. Enumeration of position types for a vertex b' and leaf d relative to b and c in T (continued).

Position-type combination $b' = a_2$ and $d = a_3$

<p>Case 9</p>	<p>Case 10</p>
<p>Case 11</p>	<p>Case 12</p>

Position-type combination $b' = a_3$ and $d = a_1$

<p>Case 13</p>	<p>Case 14</p>
--	--

Position-type combination $b' = a_3$ and $d = a_3$

<p>Case 15</p>	<p>Case 16</p>
<p>Case 17</p>	<p>Case 18</p>
<p>Case 19</p>

Theorem 2: The VIC b is an element of the path between any vertex b' and the leaf d in T that maximizes the path length $b'd$.

Proof: Proof of this theorem begins with analysis of the configurations of vertex b' and leaf d for which the path length $b'd$ can be a maximum. The statement to be proven follows as a consequence of this analysis.

Case 1

Case 1: Assume $b'd > b'v, \forall v \in T$. Then $b'd > b'c$. Now, $bc \geq bd$, for b is a VIC. But $bd = bb' + b'd > b'd$, since $bb' > 0$. Therefore $b'd < b'c$. $\otimes$.

Case 2

Case 2: Assume $b'd > b'v, \forall v \in T$. Then $b'd > b'c$. This expands to $b'\beta + \beta d > b'\beta + \beta b + bc$, or $\beta d > \beta b + bc$. But $\beta b + bc > bc \geq bd > \beta d$. $\otimes$.

Case 3

Case 3: Assume $b'd > b'v, \forall v \in T$. Then $b'd > b'c$. Then $b'b + bd > b'b + bc$, or $bd > bc$. But $bc \geq bd$, since b is a VIC. $\otimes$.

Case 4

Case 4: This case is possible, for example: Let T have only three vertices, as shown. Let $bb' = 3$, let $bc = 4$. Then $b'd = 7$, and d is the vertex farthest from b' . QED.

Case 5

Case 5: Assume $b'd > b'v, \forall v \in T$. Then $b'd > b'c$. But $bc \geq bd$ because b is a VIC. Then $b'd = b'b + bd \leq b'b + bc = b'c$. $\otimes$.

Case 6: This case is possible, for example: Let T have only four vertices as shown. Let $db = 3.5$, $bb' = 1$, and $b'c = 3$. Then b is the V1C, c is farthest from b , and d is farthest from b' . QED.

Case 7: This case is possible, for example: Let T have only 3 vertices as shown. Let $db = 3$, and $bc = 4$. Then b is the V1C, c is farthest from b , and d is farthest from $b' = c$. QED.

Case 8: Assume $b'd > b'v, \forall v \in T$. **Lemma:** $\forall v \in \text{Path } P_{bc}, v \neq b, \exists e \in T' \ni ve \geq bc$, where P_{bc} is the path between b and c , and $T' \subset T$ is a submaximal branch of T . **Proof:** Let w be the vertex of the path P_{bc} that is adjacent to b . Because w is adjacent to b , then $wq < bq \forall q \in T^*$, where $T^* \subset T$ is the maximal branch of T , and since $bw > 0$. Then $bc = bw + wc > wc$. But because b is a V1C, $\exists e \in T \ni we \geq bc$, otherwise b would not be a V1C. Vertex $e \neq b$, since $bw < bc$. Since $e \notin T^*$, $e \in T'$. Finally, $\forall v \in \text{Path } P_{bc}, v \neq w, \exists e \in T' \ni ve \geq bc$, because $ve = vw + we > we \geq bc$. QED. By this lemma, because $b' \in P_{bc}, \exists e \in T' \ni b'e \geq bc > b'c = b'd$. $\otimes$.

Case 9: Assume $b'd > b'v, \forall v \in T$. But $bc \geq bd$, because b is the V1C. Since $bb' + b'c = bc \geq bd = bb' + b'd$, then $b'c \geq b'd$. By the lemma of Case 8, $\exists e \in T' \ni b'e \geq bc > b'c \geq b'd$. $\otimes$.

Case 10

Case 10: Assume $b'd > b'v, \forall v \in T$. But $bc \geq bd$, because b is the VIC. Since $bb' + b'c = bc \geq bd = bb' + b'd$, then $b'c \geq b'd$. By the lemma of Case 8, $\exists e \in T' \ni b'e \geq bc > b'c \geq b'd$. $\otimes$.

Case 11

Case 11: Assume $b'd > b'v, \forall v \in T$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Because b is the VIC, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c$. Then $b'e = b'\beta + \beta e \geq b'\beta + bc = b'\beta + b\beta + \beta c \geq b'\beta + b\beta + \beta d = b\beta + b'd > b'd$. $\otimes$.

Case 12

Case 12: Assume $b'd > b'v, \forall v \in T$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Because b is the VIC, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c = \beta b'$. Then $b'e = b'\beta + \beta e \geq b'\beta + bc = b'\beta + b\beta + \beta c \geq b'\beta + b\beta + \beta d = b\beta + b'd > b'd$. $\otimes$.

Case 13

Case 13: This case is possible, for example: Let T have only five vertices as shown. Let $db = 4.5$, $b\beta = 1$, $\beta c = 4$, and $\beta b' = 1$. Then b is the VIC, c is farthest from b , and d is farthest from b' . QED.

Case 14

Case 14: Assume $b'd > b'v, \forall v \in T$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Then $b'e = b'\beta + \beta e \geq b'\beta + bd = b'\beta + b\beta + \beta d > b'\beta + \beta d = b'd$. $\otimes$.

Case 15

Case 15: Assume $b'd > b'v, \forall v \in T$. Because b is a V1C, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Then $b'd = b'\beta + \beta d \leq b'\beta + \beta c < b'\beta + \beta c + b\beta = b'\beta + bc \leq b'\beta + \beta e = b'e$. $\otimes$.

Case 16

Case 16: Assume $b'd > b'v, \forall v \in T$. Because b is a V1C, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c$. By the lemma of Case 8, $\exists e \in T' \ni \beta e > bc$. Then $b'd = b'\beta + \beta d \leq b'\beta + \beta c < b'\beta + \beta c + b\beta = b'\beta + bc \leq b'\beta + \beta e = b'e$. $\otimes$.

Case 17

Case 17: Assume $b'd > b'v, \forall v \in T$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Because b is a V1C, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c$. Then $b'e = b'\beta + \beta e \geq b'\beta + bc = b'\beta + b\beta + \beta c \geq b'\beta + b\beta + \beta d = b\beta + b'd > b'd$. $\otimes$.

Case 18

Case 18: Assume $b'd > b'v, \forall v \in T$. By the lemma of Case 8, $\exists e \in T' \ni \beta e \geq bc$. Because b is a V1C, $bc \geq bd$, or $b\beta + \beta c \geq b\beta + \beta d$, showing that $\beta d \leq \beta c$. Then $b'e = b'\beta + \beta e \geq b'\beta + bc \geq b'\beta + bd = b'\beta + b\beta + b'd > b'd$. $\otimes$.

Case 19

Case 19: Assume $b'd > b'v, \forall v \in T$. Because b is a V1C, $bc \geq bd$, or $b\beta_1 + \beta_1 c \geq b\beta_1 + \beta_1 d$, or $\beta_1 c \geq \beta_1 d = \beta_1 \beta_2 + \beta_2 d > \beta_2 d$. Then $b'c = b'\beta_2 + \beta_2 \beta_1 + \beta_1 c > b'\beta_2 + \beta_1 c > b'\beta_2 + \beta_2 d = b'd$. $\otimes$.

This proof shows that there are 19 configurations of a vertex $b' \neq b$ and leaf d relative to the VIC b and vertex c most distant from b in T . Of these 19 configurations, only 4 configurations are possible when leaf d is the vertex most distant from b' in T . These four configurations are described by Cases 4, 6, 7, and 13 in the enumeration of configurations shown in Table 2. These four configurations are summarized in Table 3. Case 4 is a configuration of type $b' = a_1$ and $d = a_2$. Cases 6 and 7 are configurations of type $b' = a_2$ and $d = a_1$. Case 13 is a configuration of type $b' = a_3$ and $d = a_1$. Table 4 shows two additional possible configurations if leaves d and c are equidistant from b , that is, if $bd = bc$. Note that these special cases are variants of Case 4. In every case, the VIC b is between b' and d and is an element of the path between b' and d . That is $d \notin T^*$ iff $b' \in T^*$, and $d \in T^*$ iff $b' \notin T^*$, where T^* is the maximal branch of T . QED.

Table 3. Permissible configurations of vertex b' and its most-distant leaf d relative to the VIC b and leaf c most distant from b in T .

<p>Case 4</p>	<p>Case 6</p>
<p>Case 7</p>	<p>Case 13</p>

Table 4. Additional permissible configurations of vertex b' and most-distant leaf d relative to the VIC b and leaf c most distant from b in T , if and only if leaves d and c are equidistant from b . That is, iff $bd = bc$.

<p>Case 3</p>	<p>Case 5</p>
---	--

Theorem 3: The VIC b is an element of the diameter of T .

Proof: The diameter of T is the path length ef between leaves e and f of T such that ef is a maximum over all such pairs of vertices e and f . If e is an element of the maximal branch of T , then by Theorem 2, $f=d$ and is an element of a submaximal branch of T . If e is an element of a submaximal branch of T , then $f=c$ and is an element of the maximal branch of T . Consequently, exactly one of the leaves e and f is an element of a submaximal branch of T . Let this leaf be e . Then $f=c$, and $e=d$, as d is the leaf in any

submaximal branch that maximizes the distance ec . Therefore, the diameter of T is the path length cd , and b is an element of the path between c and d since c is in the maximal branch of T while d is in a submaximal branch of T . QED. This proof confirms the result by Handler [7].

Application: Infrastructure Restoration

Applications of graph theory to the analyses of our infrastructures are important. Recent policies of the United States government emphasize the importance of infrastructure analyses [1, 2].

Consider a communications infrastructure or network consisting of a single master vertex and a set of slave vertices. The single master vertex is a command post, control center, or data-collection facility. The slave vertices are remote sensors and/or relaying stations (repeaters). This network describes many real-world communications networks, such as the supervisory control and data acquisition systems of electric-power and natural-gas utilities, military command and control systems, and data-fusion systems which combine measurements from remote sensors at a single central processing facility. The edges of this network route messages from the remote sensors to the command post, with relaying through other vertices as necessary. In general, the edges form a spanning tree selected from a set of possible communications links between the remote sensors using a rule set to optimize some feature (e.g., signal strength) of the communications infrastructure.

Consider this infrastructure network to be a strict, connected, undirected graph $G = (V, E)$ of vertices V and edges E . The master vertex will be a command post ϕ and the other vertices $v \neq \phi$ will be the slave vertices. Associate with each vertex $v \in V$ a nonnegative weight $w(v)$, called the value of v , and with each edge $e \in E$ a positive weight $l(e)$ called the length of e . The spanning-tree subgraph $T = (V, E_T)$, $E_T \subseteq E$, is a connected subgraph of G without cycles. The edges E_T consist of the active edges through which messages are routed from the slave vertices to the command post. The radius $r(v)$ of a vertex v is the distance $v\phi$ of a remote sensor v from the command-post vertex ϕ .

There are several optimal structures for T , with different spanning trees depending on the criteria for optimization. The identification of optimal spanning trees is well studied but is still a topic of recent investigation [11, 12]. A vertex one center spanning tree V1CST is an optimal structure that minimizes the largest weighted distance between the command post and the remote sensors [9]:

$$\text{V1CST} = \min_{v \in V} \max r(v). \quad (2)$$

A shortest-path tree SPT is an optimal structure that minimizes the distance between the command post and each of the remote sensors [10]:

$$\text{SPT} = \min_{v \in V} r(v). \quad (3)$$

In each of these structures, the network is a tree rooted at the command post. Figure 3 illustrates an example T . The E_T are selected from E to optimize T by some metric. In a real-world network, the optimization may be to maximize the signal-to-noise ratio of the

resulting paths, reduce the number of edges between the command post and the sensors in the longest paths, minimize the probability of data loss, or produce an extreme value for some other practical metric. Generally, the optimization can be transformed into a VICST or SPT problem by appropriate specification of the length metric $l(e)$.

Figure 3. An example spanning-tree communications infrastructure. This network T connects remote sensors and repeaters at vertices v with a command post vertex ϕ .

Much attention has been given to optimizing the network T . However, present problems in assuring infrastructure security include optimizing the restoration of the network following the failure of some network component. For the communications network G under investigation, let us turn our attention to failures that cannot be repaired and that will require reconfiguration. Reconfiguration of two detached components consists of adding a new edge with one endpoint in each component so that communication between the command post and the disconnected component is restored. The set of edges $E_T^c = \{e \in E, e \notin E_T\}$ is the complement of E_T and is the set of redundant edges available for reconfiguration of the network following a failure of a vertex or edge of the tree. A failure of an edge $e \in E_T$ will partition the tree into two detached components. An edge failure will require the addition of one new edge to reconnect the two detached components. A failure of a vertex will remove the vertex from the graph and will cause failures of the edges terminating at the failed vertex. If the vertex is a leaf, the remaining graph will have a single component, and no reconfiguration will be required. If the vertex is not a leaf, then the failure will partition the graph into λ components, where λ is the degree of the failed vertex. Then $\lambda-1$ edges must be added to the graph to reconfigure these components. Because the operation of the network depends on operation of the command-post vertex, a failure of the command post cannot be reconfigured. Figure 4 illustrates reconfiguration following an edge failure.

Figure 4. An example of reconfiguration of an infrastructure following the failure of an edge of T . The cross indicates the failed edge. A new edge e is added such that one endpoint of e is in each of the two disconnected components of T .

If G is complete and the edge distances are uniform, then the optimal network configuration is monopolar, with the command-post vertex as the monopole [8]. In this T , all vertices except the command post ϕ are leaves. Failures of edges disconnect only the leaf they connect to ϕ . Restoration consists of finding a new edge between the disconnected leaf and any other node of the graph. Let n be the number of vertices in G . Because G is complete, the degree in G of the leaf in T is $n-1$, and there will be $n-2$ edges available for reconfiguration ($n-1$ less the failed edge from E_T).

Incomplete graphs are characteristic of many real-world infrastructure networks. Examples of infrastructures composed of incomplete graphs include electric-power, natural-gas, transportation, and telephone networks. Certainly, the spanning-tree communications infrastructures under consideration are incomplete. Reconfiguring these infrastructures after the failure of a network element is less trivial than for complete graphs.

Define metrics for assessing the susceptibility of the infrastructure to failure as follows: The burden of a vertex or an edge is the sum of the values of the vertices detached from the command post by a failure of the vertex or edge. The robustness of a vertex or an edge is the sum of the maximum radius of each subgraph branch connected by the vertex or edge, divided by the maximum radius of the subgraph branch following reconfiguration. A vertex or edge is a critical component if its burden is large, if its robustness is small, or if the burden of any node following reconfiguration after a failure of the vertex or edge increases significantly. The criteria for reconfiguration could be to minimize the maximum radius of the branch through which the disconnected subgraph is reconnected or to minimize the burden of the nodes of the branch to which the disconnected subgraph is reconnected.

Consider reconfiguration of an SPT network T following an edge failure. The edge failure detaches a branch T_B from the network T . Let the criterion for

reconfiguration be to minimize the maximum radius of the reconnected branch. Then the optimal edge for reconfiguring the disconnected branch is the edge e_{ab} , with endpoint a in the component containing the command post and endpoint b in the disconnected branch such that $r_a + l(e_{ab}) + b'd$ is minimized. Here d is the leaf in the disconnected branch that is farthest from vertex b' .

This result applies directly to the problem of reconfiguring a data-fusion communications network in which the criterion for reconfiguration is to maximize the probability of successful transmission of data through the communications network. For this application, let $\rho(e)$ be the probability of successfully transmitting a message through edge $e \in E$. Then the probability $P(v)$ of successfully transmitting a message from vertex v to the command post ϕ is the product of the probabilities of successful transmission of each of the edges in the path P from v to ϕ .

$$P(v) = \prod_{e \in P} \rho(e). \quad (4)$$

Then

$$P(v) = \exp\left(\ln\left(\prod_{e \in P} \rho(e)\right)\right) = \exp\left(-\sum_{e \in P} \ln\left(\frac{1}{\rho(e)}\right)\right). \quad (5)$$

Note that $P(v)$ is maximized when $\sum_{e \in P} \ln\left(\frac{1}{\rho(e)}\right)$ is minimized. Then the optimal

reconfiguration following a failure of an element of a maximum-probability network is equivalent to the optimal reconfiguration of an SPT network where the length function $l(e)$ is the natural logarithm of the reciprocal of $\rho(e)$, $l(e) = \ln\left(\frac{1}{\rho(e)}\right)$.

The computation time to reconfigure a detached branch is reduced by examining the configurations and distances of the vertices b' and d in T_B . Because the paths between pairs of vertices in the tree subgraph T_B are unique, the vertex d and the distance $b'd$ can be determined in $O(n)$ time, where n is the number of vertices in T_B . If there are m edges in E_T^c between the two detached subgraphs that can reconfigure these two components, then an optimal reconfiguration can be determined in $O(mk)$ time. However, by determining the vertex one center b of T_B , the time required to compute the optimal reconfiguration can be reduced. The vertex one center b of a graph can be determined in $O(n^2)$ time [3]. However, since T_B is a tree, b can be determined in $O(n)$ time. When b is known, then both the vertex d and the distance $b'd$ are also known. Consequently, the optimal reconfiguration can be determined in $O(m+n)$ time, which is more efficient for most m and n .

References

- [1] President William J. Clinton, Executive Order, "Critical Infrastructure Protection, Executive Order 13010," *Federal Register* **61**, no. 138 (17 July 1996): 37347.
- [2] President William J. Clinton, Presidential Decision Directive, "Critical Infrastructure Protection: Sector Coordinators, Presidential Decision Directive 63," *Federal Register* **63**, no. 150 (5 August 1998): 41804.
- [3] S. L. Hakimi, "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," *Oper. Res.* **12**, 450–459 (1964).

- [4] S. L. Hakimi, "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems," *Oper. Res.* **13**, 462–475 (1965).
- [5] S. L. Hakimi and S. N. Maheshwari, "Optimum Locations of Centers in Networks," *Oper. Res.* **20**, 967–973 (1972).
- [6] R. Hassin and A. Tamir, "On the Minimum Diameter Spanning Tree Problem," *Info. Proc. Letters* **53**, 109–111 (1995).
- [7] G. Y. Handler, "Minimax Location of a Facility in an Undirected Tree Graph," *Transportation Science* **7**, 287–293 (1973).
- [8] J. Ho, D. T. Lee, C. Chang, and C. K. Wong, "Minimum diameter spanning trees and related problems," *SIAM J. Comput.* **20**, 987–997 (1991).
- [9] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. I: The p -Centers," *SIAM J. Appl. Math.* **37**, 513–538 (1979).
- [10] S. Khuller, B. Raghavachari, and N. Young, "Balancing Minimum Spanning Trees and Shortest-Path Trees," *Algorithmica* **14**, 305–321 (1995).
- [11] K. P. Lam and C. W. Tong, "Connectionist network for dynamic programming problems," *IEE Proceedings—Computers and Digital Techniques* **144**, 163–168 (1997).
- [12] T. Matsui, "The Minimum Spanning Tree Problem on a Planar Graph," *Disc. Appl. Math.* **58**, 91–94 (1995).
- [13] E. Minieka, "The Centers and Medians of a Graph," *Oper. Res.* **25**, 641–650 (1977).
- [14] K. R. Pruhs, "Using Local Adaptations to Reconfigure a Spanning Tree of a Network," *Disc. Appl. Math.* **57**, 67–74 (1995).
- [15] J. Spragins, *Telecommunications: Protocols and Design* (New York: Addison-Wesley, 1991).
- [16] B. C. Tansel et al., "Location on Networks: A Survey. Part I. The p -Center and p -Median Problems," *Management Science* **29**, 482–497 (1983).
- [17] B. C. Tansel et al., "Location on Networks: A Survey. Part II. Exploiting Tree Network Structure," *Management Science* **29**, 498–511 (1983).