

RESEARCH ARTICLE

Safe sets, network majority on weighted trees

Ravindra B. Bapat¹ | Shinya Fujita² | Sylvain Legay³ | Yannis Manoussakis³ | Yasuko Matsui⁴ | Tadashi Sakuma⁵ | Zsolt Tuza^{6,7}

¹Indian Statistical Institute, New Delhi, India

²International College of Arts and Sciences, Yokohama City University, Yokohama, Japan

³Laboratoire de Recherche en Informatique, University Paris-Sud, Orsay Cedex, France

⁴Department of Mathematical Sciences, Faculty of Science, Tokai University, Hiratsuka, Japan

⁵Systems Science and Information Studies, Faculty of Education, Art and Science, Yamagata University, Yamagata, Japan

⁶Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary

⁷Department of Computer Science and Systems Technology, University of Pannonia, Veszprém, Hungary

Correspondence

Tadashi Sakuma, Systems Science and Information Studies, Faculty of Education, Art and Science, Yamagata University, 1-4-12 Kojirakawa, Yamagata 990-8560, Japan.
Email: sakuma@e.yamagata-u.ac.jp

Funding information

Grant Sponsor: the JC Bose Fellowship, Department of Science and Technology, Government of India; Grant Sponsor: Scientific Research (C), Grant No.: 15K04979; Grant Sponsor: Scientific Research (C), Grant No.: 26400185; Grant Sponsor: The National Research, Development and Innovation Office – NKFIH, Grant No.: SNN116095.

Abstract

Let $G = (V, E)$ be a graph and let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of G . For every subset X of V , let $w(X) := \sum_{v \in X} w(v)$. A non-empty subset $S \subseteq V(G)$ is a *weighted safe set* if, for every component C of the subgraph induced by S and every component D of $G \setminus S$, we have $w(C) \geq w(D)$ whenever there is an edge between C and D . If the subgraph $G[S]$ induced by a weighted safe set S is connected, then the set S is called a *weighted connected safe set*. In this article, we show that the problem of computing the minimum weight of a safe set is \mathcal{NP} -hard for trees, even if the underlying tree is restricted to be a star, but it is polynomially solvable for paths. We also give an $O(n \log n)$ time 2-approximation algorithm for finding a weighted connected safe set with minimum weight in a weighted tree. Then, as a generalization of the concept of a minimum safe set, we define the concept of a parameterized infinite family of proper central subgraphs on weighted trees, whose polar ends are the vertex set of the tree and the centroid points. We show that each of these central subgraphs includes a centroid point.

KEYWORDS

approximation algorithm, centroid, network majority, \mathcal{NP} -completeness, safe set, weighted tree

1 | INTRODUCTION

We can regard a network as a mature community on a large scale; more precisely, it consists of a collection of small communities with some mutual connections. In such a network, it is important to gain control of a “majority” so that we can control the network consensus. On the other hand, for those who are concerned about network security, they would think that a network where we can easily get a majority is unstable and it has a risky structure in view of network vulnerability.

.....
An extended abstract of this paper appears in the proceedings of the 10th edition of the Jornadas de Matematica Discreta y Algoritmica (10th JMDA) [4].

Motivated by these observations, we would like to give some appropriate definition for gaining a majority in a given network. As a network model, we here consider this problem on simple undirected graphs with some given weight on each vertex. Note that each weight on a vertex represents a certain measure for importance in the network.

We use [7] for terminology and notation not defined here. Only finite, simple graphs are considered. For a graph $G = (V, E)$ and for its arbitrary vertex v , let $\deg(v)$ denote the degree of v , let $\delta(G)$ be the minimum degree of G , and let $\alpha(G)$ be the independence number of G . The order and size of G are denoted by n and m , respectively. The subgraph of G induced by a subset $S \subseteq V(G)$ is denoted by $G[S]$. When A and B are vertex-disjoint subgraphs of G , the set of edges that join some vertex of A and some vertex of B is denoted by $E(A, B)$.

Let $G = (V(G), E(G))$ be a graph and let ω be a weight function on $V(G)$ such that $\omega : V(G) \rightarrow \mathbb{R}_{>0}$. For a vertex subset S of $V(G)$, let $\omega(S) := \sum_{v \in S} \omega(v)$. We often abuse notations for vertex subsets and subgraphs. So, for a subgraph H of G , we write $\omega(H)$ for $\omega(V(H))$ (thus, $\omega(H) := \sum_{v \in V(H)} \omega(v)$).

If a connected subgraph H of G satisfies $\omega(G) \leq 2\omega(H)$ then no one may object to considering that the subnetwork H plays a majority role in G . However, one might come up with the following natural question: Do we always need to get more than half of the weight for gaining the network majority?

To answer this question, let us consider a weighted graph G with a weight function ω on $V(G)$, where we will always associate some given network \mathcal{N} with (G, ω) . (So we often identify/abuse notations (G, ω) and \mathcal{N} .) In view of graph topology, it may be natural to assume that the following three properties hold for \mathcal{N} :

- (1) For any two vertices p, q in G , any communication between p and q is conducted on a path joining p and q in G .
- (2) For a vertex subset S of G , when we consider the community associated with S in \mathcal{N} , the community S can block any communication for any two vertices in $V(G) \setminus S$ from two distinct components of $V(G) \setminus S$ by cutting off all the paths joining them.
- (3) For any two communities S_1, S_2 in G , S_1 and S_2 can form an alliance if and only if there is at least one safe way of communication (i.e., a path in which every vertex is in some community that colludes with either S_1 or S_2) between any pair of vertices in $V(S_1) \cup V(S_2)$.

For example, let us observe a weighted path $P_n = v_1 v_2 \dots v_{3n}$ with a weight function ω on $V(P)$ such that $\omega(v_i) = 1$ for all i . By taking a subpath $X = v_{n+1} v_{n+2} \dots v_{2n}$, we see that there is no component in $P \setminus V(X)$ whose weight sum exceeds the weight sum of X . Hence, under the above assumption, it would be appropriate for us to consider that X attains a majority role for any community on P . Hence we can conclude that the answer to the above question is negative. Moreover, to formulate our problem, we must consider the following basic question: How can we calculate the minimum weight of a subnetwork which attains a majority role for a given network? To answer this question, let us focus on a known concept called *safe sets*, which was introduced by Fujita, MacGillivray and Sakuma [11] for unweighted graphs. In this article, we will generalize this concept to the weighted version in a natural manner and give some basic properties along this line.

A non-empty subset $S \subseteq V(G)$ is a *safe set* if, for every component C of $G[S]$ and every component D of $G \setminus S$, we have $|C| \geq |D|$ whenever $E(C, D) \neq \emptyset$. If $G[S]$ is connected, then S is called a *connected safe set*. The minimum cardinality among all safe sets (resp. connected safe sets) of G is called the *safe number* (resp. *connected safe number*) of G and is denoted by $s(G)$ (resp. $cs(G)$). As is proven in [11], both the problem of computing the safe number and the problem of computing the connected safe number are \mathcal{NP} -hard in general while the connected safe number of a tree can be computed in linear time. Quite recently, by using dynamic programming [15], the authors in [1] obtained an $O(n^5)$ -time algorithm for finding a safe set with minimum cardinality of a tree with n vertices. By using the same method, they also proved that both the safe number $s(G)$ and the connected safe number $cs(G)$ of a given graph G of bounded treewidth can be computed in polynomial-time.

In this article, we extend this concept on graphs in which each vertex has a positive weight. Formally, let $G = (V, E)$ be a graph and let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of G . A non-empty subset $S \subseteq V(G)$ is a *weighted safe set* if, for every component C of the subgraph induced by S and every component D of $G \setminus S$, we have $w(C) \geq w(D)$ whenever $E(C, D) \neq \emptyset$. If $G[S]$ is connected, then S is called a *weighted connected safe set*. The minimum weight among all weighted safe sets (resp. connected safe sets) of (G, w) is called the *safe number* (resp. the *connected safe number*) of (G, w) and is denoted by $s(G, w)$ (resp. $cs(G, w)$).

As we mentioned before, the concept of a (weighted) safe set can be thought as a suitable measure of network vulnerability, and hence it has some clear relation to other such graph invariants. For example, the *graph integrity*, a well studied measure of reliability of a graph network (G, w) , is defined as

$$I(G) := \min_{S \subseteq V(G)} \{w(S) + \max \{w(H) : H \text{ is a component of } G[V(G) \setminus S]\}\}$$

(e.g., see [2, 3, 5, 9, 21]). From the definitions of the graph integrity and the safe number, we have the following:

Proposition 1. *For every graph network (G, w) , the inequality $I(G) \leq 2s(G, w)$ holds. Furthermore, if a set $S(\subseteq V(G))$ attains the number $I(G)$ and the induced subgraph $G[S]$ is connected, then we also have the inequality $cs(G, w) \leq I(G) \leq 2 cs(G, w)$.*

From now on, we do not consider unweighted safe sets of a weighted input graph. Hence we often omit the term “weighted” and use the abbreviation “safe set” even if the input graph is a weighted graph.

We show that a minimum safe set of weighted trees is also an appropriate indicator to express a central subgraph. In this article, we define infinitely many scalings of the concept of central subgraph, namely the α -safe sets, each of which includes a *centroid point* in a tree. A *centroid point* in a tree $T = (V, E)$ is a vertex v of T such that each weight of the connected components of the subgraph $T[V \setminus \{v\}]$ does not exceed half the weight of the tree T . In 1869, Jordan [14] defined this concept for unweighted trees, and Bielak and Pańczyk [6] generalized the definition for vertex-weighted trees in 2012. This concept has been intensively investigated in the literature [17–20, 22]. The *betweenness centrality* of a vertex (an edge) is defined as the number of shortest paths that pass through that vertex (edge). In 1977, Linton [16] defined this concept and Girvan and Newman [13] extend the definition to the case of edges. Recently the clustering of networks has received much attention and many researchers have proposed algorithms for it. Among them, some popular clustering algorithms typified by Girvan and Newman [13] tend to fail to extract communities with high betweenness centrality in a given network. (For example, some road traffic networks surely have such communities.) On the other hand, our concept of central subgraphs and algorithms to find them may be useful for extracting such communities in given networks. Note that these central subgraphs in a given unweighted tree can be found in linear time (see Remark 1 in Section 4 for more details).

The article is organized as follows.

In Section 2, we consider the time complexity of finding a minimum connected or non-connected safe set in a weighted tree. We show that this problem is \mathcal{NP} -hard even if the underlying tree is restricted to be a star. On the other hand, we construct a polynomial-time algorithm for finding a safe set with minimum weight on paths.

In Section 3, we describe an $O(n \log n)$ time algorithm to find a connected safe set of a weighted tree whose weight is at most twice the weight of a minimum safe set, that is, a 2-approximation algorithm for finding a weighted connected safe set with minimum weight in a weighted tree. Note that this algorithm can be thought as a generalization of the algorithm connected safe set in [11].

In Section 4, as a generalization of the concept of minimum safe set, we define the concept of a parameterized infinite family of proper central subgraphs on weighted trees, whose polar ends are the vertex set of the tree and the centroid points. We show that each of these central subgraphs includes a centroid point.

In Section 5, we provide conclusions and propose several open problems for future investigations.

2 | COMPLEXITY

2.1 | \mathcal{NP} -completeness of the weighted safe set problem

In this subsection, we consider the following decision problem:

CONNECTED VERTEX-WEIGHTED SAFE SET

INSTANCE: A connected graph $G = (V, E)$, a positive weight function $w : V \rightarrow \mathbb{Q}_{>0}$ on the vertex set V of G , and a positive rational number t .

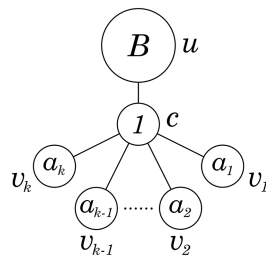


FIGURE 1 The star T and its vertex-weight function w

QUESTION: Does there exist $S \subseteq V(G)$ with $w(S) \leq t$ such that $G[S]$ is connected and $w(S) \geq w(C)$ for every component C of $G \setminus S$?

We show the \mathcal{NP} -completeness of the above problem by a reduction from the following problem:

SUBSET SUM

INSTANCE: A finite set A , a size $s(a) \in \mathbb{Z}_{>0}$ for each $a \in A$, a positive integer I .

QUESTION: Is there a subset $A^+ \subseteq A$ such that the sum of the sizes of the elements in A^+ is exactly I ?

The \mathcal{NP} -completeness of SUBSET SUM is well known.

Theorem 1 (Karp, 1972). *The problem SUBSET SUM is \mathcal{NP} -complete.*

By using the above, we derive the following:

Theorem 2. *The problem CONNECTED VERTEX-WEIGHTED SAFE SET is \mathcal{NP} -complete, even if the input graph is restricted to be a star (i.e., a tree all of whose vertices but one are leaves).*

Proof of Theorem 2. Note that CONNECTED VERTEX-WEIGHTED SAFE SET clearly belongs to the class \mathcal{NP} . Let $T = (V, E)$ be a star defined by $V = \{c, u, v_1, \dots, v_k\}$ and $E = \{cu, cv_1, \dots, cv_k\}$. Let $w : V \rightarrow \mathbb{Z}_{>0}$ be a positive integral weight function on the vertex-set V of T such that $w(c) = 1$, $w(u) = B$, $w(v_1) = a_1, \dots, w(v_k) = a_k$, and $1 + \max \{a_i | i = 1, \dots, k\} \leq B \leq \sum_{i=1}^k a_i$ hold, as indicated in the Figure 1.

The set $\{c, u\}$ is clearly a connected safe set of (T, w) . This set $\{c, u\}$ cannot be a minimum safe set of (T, w) if and only if there exists a subset $\Lambda \subseteq \{1, \dots, k\}$ such that $B - 1 = \sum_{\lambda \in \Lambda} a_\lambda$ holds. Moreover, the set $\{c, u\}$ cannot be a minimum safe set of (T, w) if and only if there exists a connected safe set whose weight is at most B . Hence, by using the above gadget, we can reduce SUBSET SUM PROBLEM to CONNECTED VERTEX-WEIGHTED SAFE SET PROBLEM in a polynomial-time, as follows:

Let $A = \{v_1, \dots, v_m\}$ be an instance of SUBSET SUM, and let $s_i := 3s(v_i)$ for each $v_i \in A$. Set $w(v_i) = s_i$ for each $v_i \in A$. Set $B := 3I + 1$. Note that $\max \{s_i | i = 1, \dots, m\} \leq 3I = B - 1$ and $3 \leq \min \{s_i | i = 1, \dots, m\}$ hold. Set $k := m + 1$ and let v_{m+1} be an element outside of A such that $w(v_{m+1}) = B - 2$. Set $V := \{u, c, v_1, \dots, v_k\}$ and $E := \{cu, cv_1, \dots, cv_k\}$. Set $w(c) := 1$, $w(u) := B$. Set $t := B$. Note that any safe set X of the pair (T, w) with $w(X) \leq B$ cannot contain the vertex v_{m+1} . Moreover we have $B \leq \sum_{i=1}^k s_i$. Hence the answer to SUBSET SUM for the instance is YES if and only if the answer to CONNECTED VERTEX-WEIGHTED SAFE SET for the instance graph $G := (V, E)$ is YES. ■

Lemma 1. *For every star $T = (V, E)$ and every positive weight function w on V , $s(T, w) = cs(T, w)$ holds.*

Proof of Lemma 1. Suppose not. Then there exists a positive weight function $w : V \rightarrow \mathbb{R}_{>0}$ such that every safe set S of (T, w) with minimum weight consists of several (at least two) leaves of T and $T[V \setminus S]$ is connected. Let u, v be two elements of S . Without loss of generality, we assume that $w(u) \leq w(v)$. Since $w(V \setminus S) \leq w(u)$, $w(V \setminus (S \setminus \{u\})) \leq w(S)$ holds. Since the set $S' := V \setminus (S \setminus \{u\})$ contains the central vertex of T , $T[S']$ is connected. Since $v \in S'$, we have $w(u) \leq w(S)$. Thus S' is a safe set whose weight is less than $w(S)$, which is a contradiction. ■

Combining Theorem 2 and Lemma 1, we have the following:

Corollary 1. *The problem of computing a safe set with minimum weight is \mathcal{NP} -hard even if the input graph is restricted to be a star.*

Actually, the problem CONNECTED VERTEX-WEIGHTED SAFE SET is \mathcal{NP} -complete on the following large class of graphs:

Corollary 2. *For an arbitrary given connected graph H , we have the following: The problem CONNECTED VERTEX-WEIGHTED SAFE SET is \mathcal{NP} -complete even if the input graph G is restricted to have a bridge e such that $G - e$ is a disjoint union of a star and the graph H .*

Proof of Corollary 2. Let all of V, E and w be as defined in the last paragraph of the above proof of Theorem 2. Let h be a vertex of the graph H . Set $V' := V \cup V(H)$ and set $E' := E \cup E(H) \cup \{hv_{m+1}\}$. Reset $w(v_{m+1}) := (B-2)-0.1$. For every element v of V , set $w'(v) := w(v)$. For every vertex q of the graph H , set $w'(q) := \frac{1}{10|V(H)|}$. Let $G := (V', E')$. Then w' is a positive weight function on the vertex-set of G such that the pair (T, w) has a safe set S with $w(S) \leq B$ if and only if the pair (G, w') has a safe set S' with $w(S') \leq B$. Hence the proof is complete. ■

2.2 | Weighted safe set of paths

In this subsection, we consider the following optimization problem on paths.

WEIGHTED SAFE SET OF PATHS

INPUT: A path graph $P = (V, E)$ such that $|V| \geq 3$, and a positive weight function $w : V \rightarrow \mathbb{Q}_{>0}$.

OUTPUT: A safe set $S \subseteq V$ with minimum weight on P .

Theorem 3. *Finding a safe set with minimum weight is polynomial-time solvable on paths. Our algorithm requires $O(n^3)$ time and space, where n is the number of vertices of a given path graph $P = (V, E)$.*

To prove this theorem, we show that the problem of finding a weighted safe set of a path is equivalent to finding a shortest weighted path on the acyclic digraph defined as follows: Let P be a path of n vertices v_1, v_2, \dots, v_n , with positive weights w_1, w_2, \dots, w_n , respectively. For $1 \leq i \leq j \leq n$, we call $P_{i,j}$ the subpath of P consisting of the vertices v_i, v_{i+1}, \dots, v_j .

From P , we will construct the weighted digraph $G_P = (V(G_P), A(G_P))$ with the weight function ω on $V(P)$ as follows:

$$\begin{aligned} V(G_P) &= \{u_{i,j}, v_{i,j} | 1 \leq i \leq j \leq n, (i,j) \neq (1,n)\} \cup \{t_0, t_\infty\}, \\ A(G_P) &= \left\{ \begin{aligned} &\{(u_{i,j}, v_{j+1,k}) | 1 \leq i \leq j < k \leq n, w(P_{i,j}) \geq w(P_{j+1,k})\} \cup \\ &\{(v_{i,j}, u_{j+1,k}) | 1 \leq i \leq j < k \leq n, w(P_{i,j}) \leq w(P_{j+1,k})\} \cup \\ &\{(t_0, u_{1,j}), (t_0, v_{1,j}) | j \in \{1, 2, \dots, n-1\}\} \cup \\ &\{(u_{i,n}, t_\infty), (v_{i,n}, t_\infty) | i \in \{2, \dots, n\}\}, \end{aligned} \right. \\ &\forall i, j, w(u_{i,j}) = w(P_{i,j}), w(v_{i,j}) = 0, \\ &w(t_0) = w(t_\infty) = 0. \end{aligned}$$

Lemma 2. *There exists a bijection between the safe sets of P and the $t_0 - t_\infty$ paths in G_P .*

Proof of Lemma 2. Let Q be any directed $t_0 - t_\infty$ path in G_P . Q can be described by a set of pairs $\{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\}$ such that Q is the path $t_0, \dots, v_{j_{l-1}, i_{l-1}}, u_{i_l, j_l}, v_{j_{l+1}, i_{l+1}}, \dots, t_\infty$. The fact that there is a directed edge

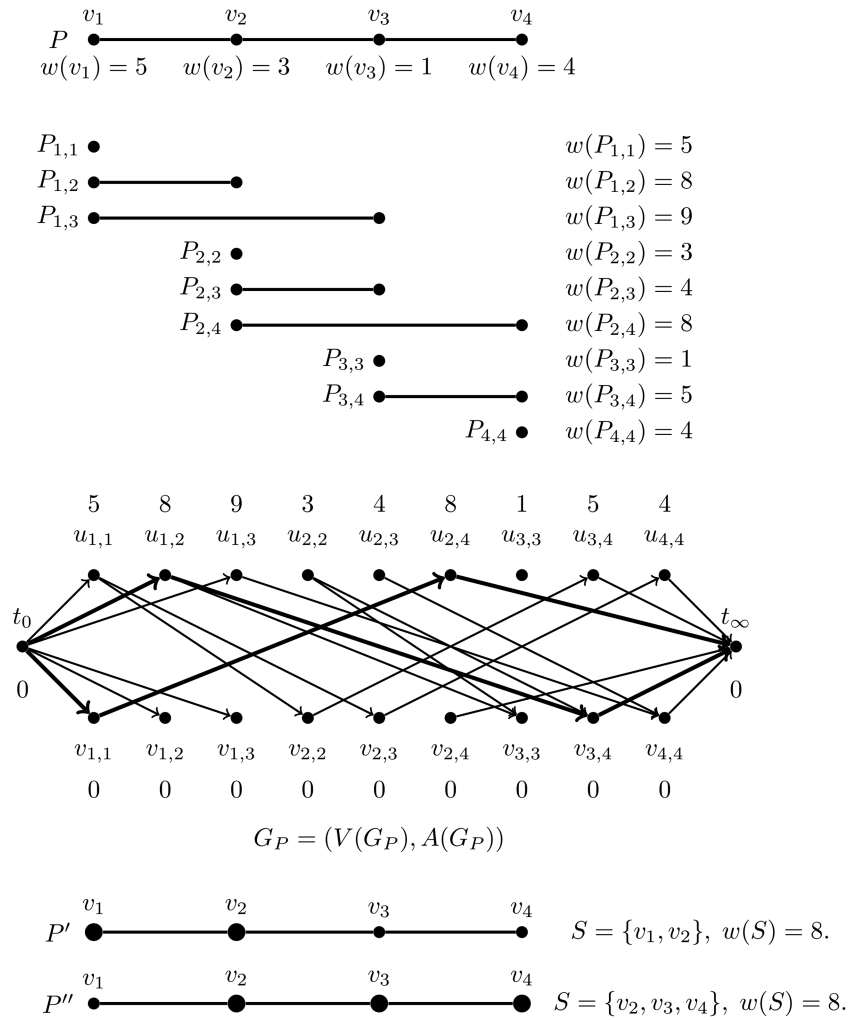


FIGURE 2 A path P, P_{ij} and the corresponding directed graph G_P . Each shortest $t_0 - t_\infty$ path in G_P corresponds to a minimum safe set S of P , and vice versa

between $v_{j_{l-1}+1, i_{l-1}}$ and u_{i_l, j_l} implies that $w(P_{j_{l-1}+1, i_{l-1}}) \leq w(P_{i_l, j_l})$ and the fact that there is a directed edge between u_{i_l, j_l} and $v_{j_{l+1}, i_{l+1}-1}$ implies that $w(P_{i_l, j_l}) \geq w(P_{j_{l+1}, i_{l+1}-1})$. These conditions satisfy the definition of components C of $G[S]$ and D of $G \setminus S$ weighted safe set S . Then, $\cup_{l=1}^k P_{i_l, j_l}$ is a weighted safe set of P . ■

In this correspondence, a safe set of P is composed of components of the form P_{i_j} , and the property of being a safe set is translated to the condition that these components come from a directed path in G_P as described above.

Lemma 3. *The weight of a safe set of P is equal to the weight of its $t_0 - t_\infty$ path in G_P .*

The above lemma is obvious from the configuration of G_P . A small example is shown in Figure 2.

Lemma 4. *For a given path P , we can construct the weighted directed graph $G_P = (V(G_P), A(G_P))$ in $O(n^3)$ time and find a minimum $t_0 - t_\infty$ path of G in $O(n^3)$ time, where n is the number of vertices of P .*

Proof of Lemma 4. For a given path $P = (V, E)$, let n be the number of vertices in V . To construct the corresponding directed graph $G_P = (V(G_P), A(G_P))$, we first construct all P_{i_j} and calculate $w(P_{i_j})$. The number of P_{i_j} s, which is same as the number of u_{i_j} s or v_{i_j} s, is $\sum_{i=1}^{|V|} i - 1 = \frac{n(n+1)}{2} - 1$. Next, we prepare a set of vertices of $V(G_P)$ which consists of all u_{i_j}, v_{i_j}, t_0 and t_∞ . Then, $|V(G_P)| = 2 \times (\frac{n(n+1)}{2} - 1) + 2 = n(n+1)$. For each vertex $v \in V(G_P)$, its degree is at most n . Thus we have $|A(G_P)| = O(n^3)$ and hence the size of G_P

is $O(n^3)$. Lastly, we find a minimum $t_0 - t_\infty$ path of G_P by using Dijkstra's algorithm [8]. The running time is $O(|V(G_P)| \log |V(G_P)|) = O(n^2 \log n)$. Hence, our algorithm requires $O(n^3)$ time and space. ■

Proof of Theorem 3. The statement is clearly derived from Lemmas 1, 2, and 3. ■

By using Eppstein's algorithm [10], we can also enumerate all the paths of minimum vertex-weight in G_P . This yields the following:

Corollary 3. *All safe sets with minimum weight can be enumerated in $O(n^3 + k)$ time on paths, where n is the number of vertices of a given path graph $P = (V, E)$ and k is the number of safe sets with minimum weight. The delay between two consecutive outputs is $O(1)$ time.*

3 | 2-APPROXIMATION ALGORITHM ON TREES

In this section we describe an $O(n \log n)$ time algorithm to find a connected safe set of a weighted tree whose weight is at most twice the weight of a minimum safe set, that is, a 2-approximation algorithm for finding a weighted connected safe set with minimum weight in a weighted tree.

Let $G = (V, E)$ be a graph of order n . For each subset X of V , let $N_G(X)$ denote the *open neighborhood* $\{y \in V \setminus X \mid \exists x \in X \text{ such that } xy \in E(G)\}$ of X . We will omit the subscript G and simply write $N(X)$ whenever no confusion arises.

CONNECTED WEIGHTED α -SAFE SET (CW α S)

INPUT: A tree $T = (V, E)$ with at least two vertices, a positive weight function $w : V \rightarrow \mathbb{Q}_{>0}$ on the vertices of T and a non-negative real number α

OUTPUT: A minimal set S on condition that every component X of $T[V \setminus S]$ satisfies $w(V(X)) \leq \alpha \cdot w(S)$

INITIALIZATION

Find the set L of leaves of T

Set $S = V(T)$

$T[S]$ is always connected

Set $t(v) = \deg(v)$ for all $v \in S$

$t(v) = |N(v) \cap S|$

Set $Bdy(S) = L$

$v \in Bdy(S) \Leftrightarrow t(v) = 1 \Leftrightarrow v$ has only one neighbor in S

Set $c(x) = \begin{cases} w(x) & x \in Bdy(S) \\ 0 & \text{otherwise} \end{cases}$

weight of the component arising from removing x

Set u to be a leaf of T such that $w(u) = \min \{w(p) \mid p \in L\}$ holds

For every vertex set $X (\subseteq V)$ such that $T[X]$ is connected, let $f(X) := \begin{cases} \alpha \cdot w(X) + \frac{(1-\alpha) \cdot w(V)}{2} & 0 \leq \alpha \leq 1 \\ \frac{1}{\alpha} \cdot w(X) & 1 \leq \alpha \end{cases}$

MAIN LOOP

While $c(u) \leq f(S \setminus \{u\})$ Do

 Remove u from S , and from $Bdy(S)$

 Let v be the unique vertex $N(u) \cap S$

 Decrement $t(v)$

 If $t(v) = 1$, then

 Insert v into $Bdy(S)$

 Set $c(v) := w(v) + \sum_{p \in N_T(v)} c(p)$

 Set $U := \{x \in Bdy(S) \mid c(x) = \min \{c(p) \mid p \in Bdy(S)\}\}$

 Set u to be a vertex in U whose weight satisfies $w(u) = \min \{w(p) \mid p \in U\}$

RETURN S .

Let us define $\delta(T, w) := \min \{w(X) - w(Y) \mid X, Y \subseteq V(T), w(X) > w(Y)\}$ and $\Delta_\alpha := \begin{cases} \alpha \cdot \delta(T, w) & 0 \leq \alpha \leq 1 \\ \frac{1}{\alpha} \cdot \delta(T, w) & 1 \leq \alpha \end{cases}$. Then we have the following:

Lemma 5. *Let S be an output of Algorithm $CW\alpha S$ for the input triple (T, w, α) . If $f(S) - \max \{c(x) \mid x \in V \setminus S\} \leq \Delta_\alpha$ holds, then S has the minimum weight $w(S)$ on condition that $T[S]$ is connected and that $\max \{c(x) \mid x \in V \setminus S\} \leq f(S)$ holds.*

Proof of Lemma 5. Let us prove the lemma by reductio ad absurdum. If the statement of the lemma does not hold, then there exists a connected subgraph X of T such that $w(V(X)) < w(S) - \delta(T, w)$ (and hence $f(V(X)) < f(S) - \Delta_\alpha$), and every component Y of $T[V \setminus V(X)]$ satisfies $w(V(Y)) \leq f(V(X))$. On the other hand, by assumption, there exists a connected component Z of $T[V \setminus S]$ such that $f(S) - \Delta_\alpha \leq w(V(Z)) \leq f(S)$. Let u be a unique vertex of Z such that $T[S \cup \{u\}]$ is connected. There exists at least one connected component C of $T[V \setminus V(X)]$ such that $V(C) \cap S \neq \emptyset$. Let p be a leaf of $T[S]$ such that $p \in V(C) \cap S$. Let P be the connected component of $T[V \setminus (S \setminus \{p\})]$ containing p . Since $c(p) = w(V(P)) \leq w(V(C)) \leq f(V(X)) < f(S) - \Delta_\alpha \leq w(V(Z)) = c(u)$, according to the instructions of Algorithm $CW\alpha S$, the vertex p must be removed from S before the vertex u is removed from S , which is a contradiction. This proves the lemma. ■

Theorem 4. *The following statements hold.*

- (1) *The running time of Algorithm $CW\alpha S$ is $O(n \log n)$.*
- (2) *Let S be an output of Algorithm $CW\alpha S$ for the triple $(T, w, 1)$. Then we have $w(S) \leq 2 cs(T, w)$.*

Proof of Theorem 4. First we will prove (1). We will sort the set $\{(v, c(v)) \mid v \in \text{Bdy}(S)\}$ in ascending order of values of $c(v)$ and, for the elements which have the same value of $c(v)$, next we will sort them in ascending order of values of $w(v)$. Clearly, we can carry out this sort to the initial set $\{(v, c(v)) \mid v \text{ is a leaf of } T\}$ and make the ordered list in $O(n \log n)$ time. For a given new element $(x, c(x))$ to be added in the ordered list, we can insert it into the correct place in the list in $O(\log n)$ time. By using this maintained list, we can perform all other steps in the loop in $O(1)$ time. Since the main loop is executed at most n times, the total time-complexity of the algorithm is $O(n \log n)$. This proves (1).

Next we will show the proof of (2). Let r be an arbitrary vertex in the set $\{x \in \text{Bdy}(S) \mid c(x) - w(x) = \min \{c(p) - w(p) \mid p \in \text{Bdy}(S)\}\}$.

We divide the proof into the following two cases.

Case 1. $w(S) \leq c(r) - w(r)$ holds.

In this case, every leaf p of $T[S]$ satisfies $w(S) \leq c(p) - w(p)$. Then let us prove the theorem by reductio ad absurdum.

Suppose that there exists a connected safe set X of (T, w) such that $w(X) < w(S)$. Then there exists a leaf u of $T[S]$ such that $u \in S \setminus X$. Let v be a unique vertex of $T[S]$ adjacent to u . Let Y be a subset of V such that $T[Y]$ is a connected component of $T[V \setminus (S \setminus \{u\})]$ containing u . And let Z be a subset of V such that $T[Z]$ is a connected component of $T[V \setminus X]$ containing u . It is clear that $X \cap S \neq \emptyset$, and hence $Y \subseteq Z$. Then we have $w(S) \leq c(u) - w(u) < c(u) = w(Y) \leq w(Z) \leq w(X)$, which is a contradiction. Hence S is a minimum connected safe set of (T, w) . This proves (2) in Case 1.

Case 2. $w(S) > c(r) - w(r)$ holds.

Let ϵ be a solution of the equation $(w(S) - \epsilon \cdot w(r)) = c(r) - (1 - \epsilon)w(r)$. Since $w(S) > c(r) - w(r)$ and $(w(S) - w(r)) < c(r)$, this ϵ satisfies $0 < \epsilon < 1$.

Let rt be the unique edge of $T[S]$ adjacent to the vertex r , and let $T^+ := (V^+, E^+)$ be a new graph resulting from a subdivision of the edge rt of T defined as follows:

$$V^+ := (V(T) \setminus \{r\}) \cup \{r_1, r_2\}$$

$$E^+ := \{pq \in E(T) \mid r \notin \{p, q\}\} \cup \{pr_1 \mid p \neq t \text{ and } pr \in E(T)\} \cup \{r_1r_2, r_2t\}$$

Let $w^+ : V^+ \rightarrow \mathbb{Q}_{>0}$ be a positive weight function on the vertices of T^+ defined as follows:

$$w^+(x) := \begin{cases} w(x) & x \in V^+ \setminus \{r_1, r_2\} (= V \setminus \{r\}) \\ \epsilon \cdot w(r) & x = r_1 \\ (1 - \epsilon) \cdot w(r) & x = r_2 \end{cases}$$

Let A be an arbitrary connected safe set of the pair (T, w) and let A^+ be the subset of V^+ defined as follows:

$$A^+ := \begin{cases} A & r \notin A \\ (A \setminus \{r\}) \cup \{r_1, r_2\} & r \in A \end{cases}$$

It is easy to see that A^+ is a safe set of (T^+, w^+) and $w(A) = w^+(A^+)$. Hence we have $cs(T^+, w^+) \leq cs(T, w)$.

Now let S' denote the subset $(S \setminus \{r\}) \cup \{r_2\}$ of V^+ . By the assumption of Case 2, the set S' is a connected safe set of the pair (T^+, w^+) . Furthermore, according to Lemma 5, the weight $w^+(S')$ is minimum among all the safe sets of (T^+, w^+) , that is, the set S' is a minimum connected safe set of (T^+, w^+) . Hence we have $w^+(S') = w(S) - \epsilon \cdot w(r) \leq cs(T, w) \leq w(S)$. On the other hand, by definition of ϵ , we have $\epsilon \cdot w(r) \leq (c(r) - w(r)) + \epsilon \cdot w(r) = w(S) - \epsilon \cdot w(r)$. And hence $w(S) \leq w(S) + (w(S) - 2\epsilon \cdot w(r)) = 2(w(S) - \epsilon \cdot w(r)) \leq 2 cs(T, w)$, which proves (2) in Case 2. ■

The following lemma is a direct generalization of Proposition 2 of [11]. We omit the proof of this lemma since it is exactly the same as the original one except for replacing cardinalities with weights.

Lemma 6. *Let $G = (V, E)$ be a connected graph and let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of G . Then $s(G, w) \leq cs(G, w) < 2s(G, w)$.*

Combining Lemma 6 and Theorem 4, we have the following:

Corollary 4. *There exists an $O(n \log n)$ time 4-approximation algorithm for finding a weighted safe set with minimum weight in a weighted tree.*

4 | CENTROID AND ITS GENERALIZATION

In this section we deal with a generalization of both centroid points and connected safe sets of weighted trees.

Let $T = (V, E)$ be a tree and let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of T . For every $\alpha \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and for every vertex set $X (\subseteq V)$ such that $T[X]$ is connected, let us define

$$f(\alpha, X) := \begin{cases} \alpha \cdot w(X) + \frac{(1-\alpha) \cdot w(V)}{2} & 0 \leq \alpha \leq 1 \\ \frac{1}{\alpha} \cdot w(X) & 1 \leq \alpha \end{cases}.$$

Then let

$$F_\alpha(T, w) := \{X \subseteq V \mid T[X] \text{ is connected and } T[V \setminus X] \text{ has no component whose weight exceeds } f(\alpha, X)\},$$

$s_\alpha(T, w) := \min \{w(X) \mid X \in F_\alpha(T, w)\}$ and let $F^{\min_\alpha}(T, w) := \{X \in F_\alpha(T, w) \mid w(X) = s_\alpha(T, w)\}$. It is clear that $F^{\min_\infty}(T, w) = \{V\}$. A member of $F^{\min_1}(T, w)$ is a *minimum safe set* of (T, w) , while every member of $F^{\min_0}(T, w)$ consists of exactly one vertex, which is called a *centroid point* of (T, w) . For any $\alpha \geq 0$, let us call a member of $F_\alpha(T, w)$ an α -safe set of the pair (T, w) .

First we will see that every α -safe set is also a β -safe set if $0 \leq \beta \leq \alpha$.

Proposition 2. *If $0 \leq \beta \leq \alpha$, then $F_\beta(T, w) \supseteq F_\alpha(T, w)$.*

Proof of Proposition 2. In the case of $1 \leq \beta \leq \alpha$, the statement clearly holds by definition of $F_\alpha(T, w)$.

Next, let us prove in the case of $0 \leq \beta \leq \alpha \leq 1$. Let X be an arbitrary set in $F_\alpha(T, w)$.

If $w(X) \geq \frac{w(V)}{2}$, then $\beta \cdot w(X) + \frac{(1-\beta) \cdot w(V)}{2} \geq \frac{w(V)}{2} \geq w(V \setminus X)$. Since the weight of any component of $T[V \setminus X]$ does not exceed $w(V \setminus X)$, by the definition of $F_\beta(T, w)$, X is in $F_\beta(T, w)$.

If $w(X) < \frac{w(V)}{2}$, then $\beta \cdot w(X) + \frac{(1-\beta) \cdot w(V)}{2} \geq \alpha \cdot w(X) + \frac{(1-\alpha) \cdot w(V)}{2}$, and hence, again by the definition of $F_\beta(T, w)$, X is in $F_\beta(T, w)$.

Combining the above two cases, we have $F_\beta(T, w) \supseteq F_1(T, w) \supseteq F_\alpha(T, w)$ for the case of $0 \leq \beta \leq 1 \leq \alpha$. ■

Corollary 5. *If $0 \leq \beta \leq \alpha$, then $s_\beta(T, w) \leq s_\alpha(T, w)$.*

Corollary 6. *For every real number $\alpha \in [0, 1]$, $s_\alpha(T, w) \leq s_1(T, w) \leq \frac{w(V)}{2}$.*

Lemma 7 ([6]). *Let $T = (V, E)$ be a tree with n vertices, and let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of T . If the pair (T, w) has at least two distinct centroid points $u, v \in V$, then uv is an edge of T and the set $\{u, v\}$ is the set of all centroid points of T .*

Proof of Lemma 7. Let P denote the path from u to v on T . Let p be the vertex on P adjacent to u , and let q be the vertex on P adjacent to v . Also let T_p denote the component of $T - uv$ containing p and v , and let T_q be the component of $T - uv$ containing q and u . Since u and v are centroid points of T , both $w(V(T_p)) \leq \frac{w(V)}{2}$ and $w(V(T_q)) \leq \frac{w(V)}{2}$. If $V(P) \setminus \{u, v\} \neq \emptyset$, then $V = V(T_p) \cup V(T_q)$ and $V(P) \setminus \{u, v\} \subseteq V(T_p) \cap V(T_q)$ and $w(V(P) \setminus \{u, v\}) > 0$. Hence, we have $w(V) < w(V(T_p)) + w(V(T_q)) \leq w(V)$, which is a contradiction. ■

Every minimum α -safe set contains a centroid point. More precisely we have:

Proposition 3. *Let $T = (V, E)$ be a tree with n vertices, let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of T , and let α be an arbitrary positive real number. Let u be a centroid point of (T, w) . If an element S in $F^{\min_\alpha}(T, w)$ does not contain the centroid point u , then the pair (T, w) has exactly two centroid points and S must contain the other centroid point v , $w(S) = \frac{w(V)}{2}$ holds, and $V \setminus S$ is also an element of $F_\alpha^{\min}(T, w)$. Furthermore, in this case, $\{S, V \setminus S\} \subseteq F_1^{\min}(T, w)$ and $s_\alpha(T, w) = s_1(T, w)$.*

Proof of Proposition 3. Let u be a centroid point of (T, w) . Suppose that an element S in $F^{\min_\alpha}(T, w)$ does not contain the centroid point u . Suppose that S does not contain any centroid point of (T, w) . Then the maximum weight among all components of $T[V \setminus S]$ is strictly more than $\frac{w(V)}{2}$. By the definition of $f(\alpha, S)$, we have $\alpha < 1$. And hence, according to Corollary 6, $s_\alpha(T, w) \leq \frac{w(V)}{2}$. Then, by the definition of $f(\alpha, S)$, we have $f(\alpha, S) \leq \frac{w(V)}{2}$. On the other hand, by the definitions of $F_\alpha(T, w)$ and $F^{\min_\alpha}(T, w)$, no component of $T[V \setminus S]$ can have weight larger than $f(\alpha, S)$, which is a contradiction. Hence S contains a centroid point v . In this case, from Lemma 7, the pair $\{u, v\}$ is an edge of T . Thus $T - uv$ has two components, each of whose weights is equal to $\frac{w(V)}{2}$. It means that $\alpha \cdot w(S) + \frac{(1-\alpha) \cdot w(V)}{2} = \frac{w(V)}{2}$, and hence either $\alpha = 0$ or $w(S) = \frac{w(V)}{2}$. Since $\alpha > 0$, we have $s_\alpha(T, w) = w(S) = \frac{w(V)}{2}$. Thus $s_\alpha(T, w) = s_1(T, w)$. ■

Corollary 7. *Let $T = (V, E)$ be a tree with n vertices, let $w : V \rightarrow \mathbb{R}_{>0}$ be a positive weight function on the vertices of T , and let α, β be positive real numbers. If $s_\alpha(T, w) < \frac{w(V)}{2}$ and $\alpha < \beta$, then, for every member S of $F_\beta(T, w)$, S contains all the centroid points of (T, w) .*

Remark 1. Let T be an unweighted tree (i.e., $\forall v \in V(T), w(v) = 1$). In this case we have $\delta(T, w) = 1$ and hence the assumption in Lemma 5 always holds for every non-negative real number α . This means that Algorithm CW_αS in Section 3 finds a minimum α -safe set of T correctly. Moreover, since, for every subset X of $V(T)$, its weight $w(X)$ is in $\{1, \dots, |V(T)|\}$, we can omit the sorting phase in Algorithm CW_αS by using an array of length $|V(T)|$. Hence we can find a minimum α -safe set of T in linear time (in the case of $\alpha = 1$, this observation is exactly the same as Theorem 2 in [11]).

5 | CONCLUSION

We have proposed some majority concepts for networks, which are generalizations of the “safe set” and the “connected safe set” in [11] to weighted graphs. We have shown that the problem of minimizing these objectives is \mathcal{NP} -hard, even if the input graph is restricted to be a star, while it is polynomially solvable for paths. We have given polynomial-time approximation algorithms for them. Lastly, as a further generalization of these concepts, we have defined the concept of a parameterized infinite family of proper central subgraphs on weighted trees. We have shown that each of these central subgraphs includes a centroid point.

Finally, we give some open problems for future investigations.

1. How should we generalize the concept of α -safe set to the case of a general graph $G = (V, E)$? Since general graphs do not necessarily have centroid points, it may not be appropriate to define $f(\alpha, X)$ as $\alpha \cdot w(X) + \frac{(1-\alpha) \cdot w(V)}{2}$ for $0 \leq \alpha \leq 1$. Maybe, the simplest way is to define it as $f(\alpha, X) := \frac{1}{\alpha} \cdot w(X)$. Also in this version of an α -safe set, $F^{\min\infty}(G, w) = \{V\}$ holds and $F^{\min 1}(G, w)$ is again the set of the minimum safe sets of (G, w) . It is doubtful whether we can say that this version of an α -safe set S is a central part of the given weighted graph G in the case when $\frac{w(V)}{2} < f(\alpha, S)$. As a matter of fact, under this definition, we have $F^{\min 0}(G, w) = \{\emptyset\}$.
2. There are several generalizations of the centroids of trees to the case of general graphs. It would be interesting if one can make clear the relations between these generalizations and the (connected) safe sets.
3. Do there exist more accurate approximation algorithms for finding a weighted (connected) safe set with minimum weight in a weighted tree? In particular, the approximation ratio in Corollary 4 is very likely far from the optimal. It would be more preferable if one can find a polynomial-time approximation scheme (PTAS) computing a (connected) safe set with minimum weight of a weighted graph of bounded treewidth. Although a pseudo-polynomial-time algorithm is easily obtained from the polynomial-time algorithm for the cardinality version in [1], it seems to be difficult to modify this pseudo-polynomial-time algorithm to be a PTAS.
4. It would be worth characterizing the class of graphs $G = (V, E)$ such that $cs(G, w) = s(G, w)$ holds for any positive weight function w on V . As mentioned in Lemma 1, the stars $T = (V, E)$ belong to this class. In fact there is ongoing research on this problem [12]. Details will be discussed elsewhere.

ACKNOWLEDGMENTS

The first author gratefully acknowledges support from the JC Bose Fellowship, Department of Science and Technology, Government of India. The second author’s research is supported by Grant-in-Aid for Scientific Research (C) (15K04979). The sixth author’s research is supported by Grant-in-Aid for Scientific Research (C) (26400185). The seventh author’s research is supported by the National Research, Development and Innovation Office – NKFIH under the grant SNN116095.

REFERENCES

- [1] R. Águeda et al., Safe sets in graphs: Graph classes and structural parameters, *Combinatorial Optimization and Applications. COCOA 2016. Lecture Notes in Computer Science*, (T.H. Chan, M. Li, L. Wang, eds.), Springer, Cham, 2016.
- [2] M. Atici and C. Ernst, *On the range of possible integrities of graphs $G(n, k)$* , *Graphs Comb.* **27** (2011), 475–485.
- [3] K.S. Bagga et al., *A survey of integrity*, *Discrete Appl. Math.* **37/38** (1992), 13–28.
- [4] R.B. Bapat et al., *Network majority on tree topological network*, *Proceedings of the 10th edition of the Jornadas de Matematica Discreta y Algoritmica (the 10th JMDA)*, *Electron. Notes Discrete Math.* **54** (2016), 79–84.
- [5] D. Benko, C. Ernst, and D. Lanphier, *Asymptotic bounds on the integrity of graphs and separator theorems for graphs*, *SIAM J. Discrete Math.* **23** (2009), 265–277.
- [6] H. Bielak and M. Pańczyk, *A self-stabilizing algorithm for finding weighted centroid in trees*, *Ann. UMCS Informatica AI XII* **2** (2012), 27–37.
- [7] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs and digraphs* (5th ed.), Chapman and Hall, London, 2011.
- [8] E.W. Dijkstra, *A note on two problems in connexion with graphs*, *Numer. Math.* **1** (1959), 269–271.
- [9] P.G. Drange, M.S. Dregi, and P. van’t Hof, *On the computational complexity of vertex integrity and component order connectivity*, *Algorithms and Computation*, Proc. 25th Int. Symp. ISAAC 2014, *Lecture Notes in Computer Science*, vol. **8889**, Springer, Berlin, 2014, pp. 285–297.
- [10] D. Eppstein, *Finding the k shortest paths*, *SIAM J. Comput.* **28** (1998), 652–673.
- [11] S. Fujita, G. MacGillivray, and T. Sakuma, *Safe set problem on graphs*, *Discrete Appl. Math.* **215** (2016), 106–111.
- [12] S. Fujita, B. Park, and T. Sakuma, Personal communication.

- [13] M. Girvan and M.E.J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA **99** (2002), 7821–7826.
- [14] C. Jordan, *Sur les assemblages de lignes*, J. Reine Angew. Math. **70** (1869), 185–190.
- [15] B. Korte and J. Vygen, *Combinatorial optimization* (5th ed.), Springer, Berlin, 2011.
- [16] F. Linton, *A set of measures of centrality based on betweenness*, Sociometry **40** (1977), 35–41.
- [17] W. Piotrowski and M.N. Sysło, *Some properties of graph centroids*, Ann. Oper. Res. **33** (1991), 227–236.
- [18] P.J. Slater, *Centers to centroids in graphs*, J. Graph Theory **2** (1978), 209–222.
- [19] P.J. Slater, *Accretion centroids: A generalization of branch weight centroids*, Discrete Appl. Math. **3** (1981), 187–192.
- [20] C. Smart and P.J. Slater, *Center, median, and centroid subgraphs*, Networks **34** (1999), 303–311.
- [21] A. Vince, *The integrity of a cubic graph*, Discrete Appl. Math. **140** (2004), 223–239.
- [22] H. Wang, *Centroid, leaf-centroid, and internal-centroid*, Graphs Comb. **31** (2015), 783–793.

How to cite this article: Bapat RB, Fujita S, Legay S, et al. Safe sets, network majority on weighted trees. *Networks*. 2018;71:81–92. <https://doi.org/10.1002/net.21794>