

FINDING SAFE STRATEGIES FOR COMPETITIVE DIFFUSION ON TREES

Jeannette Janssen and Celeste Vautour

Department of Mathematics and Statistics, Dalhousie University, Halifax, Nova Scotia, Canada

Abstract *We study the two-player safe game of Competitive Diffusion, a game-theoretic model for the diffusion of technologies or influence through a social network. In game theory, safe strategies are mixed strategies with a minimum expected gain against unknown strategies of the opponents. Safe strategies for competitive diffusion lead to maximum spread of influence in the presence of uncertainty about the other players. We study the safe game on two specific classes of trees, spiders and complete trees, and give tight bounds on the minimum expected gain. We then use these results to give an algorithm that suggests a safe strategy for a player on any tree. We test this algorithm on randomly generated trees and show that it finds strategies that are close to optimal.*

1. INTRODUCTION

Online social networks such as Facebook, Twitter, and LinkedIn have an increasingly important role in the spread of information through society. News about all kind of topics can spread quickly along the “friend” or “follower” links in the network. Understanding and modeling this process, and determining best strategies for reaching a large number of users, is instrumental for commercial applications such as viral marketing, but also for social activism and societal benefit, such as countering false rumors, spreading information about safe health practices, etc.

Competitive Diffusion, which was introduced in [1] is a game-theoretic model for the diffusion of information in a network. This game is built on the assumption that there are several players, who wish to spread competing information. One can think of companies encouraging consumers to adopt their products, or political organizations spreading a point of view about a contentious issue. The goal of each player is to reach the largest possible number of users. The messages spread by the players are assumed to be competitive, so any user who adopts the view of one of the players will not be susceptible to the messages sent by the other players. Moreover, users adopt the view of the player whose message is the first to reach them. If two competing messages reach the user at the same time, the user adopts a neutral position and effectively blocks the passage of information.

Competitive diffusion lends itself to analysis via game theory. Because of the possibilities that users turn “neutral,” it is not a zero-sum game. In this study, we consider

the associated safe game. This game focuses on one particular player, here referred to as Player 1, and the aim is to maximize the minimum gain of Player 1, regardless of the strategies of the other players. The safe game can be interpreted as the game in which the goal of all other players is to minimize the gain of Player 1, rather than to in which the goal of maximize their own gain. The reason to adopt the safe game scenario is because the traditional game assumes full information about the strategies of the other player. The safe game explores the scenario where in the strategies of the other players are unknown, thus, the safest scenario for Player 1 is to assume that the other players are actively countering his/her strategy. This contrasts with the analysis of competitive diffusion in terms of pure Nash equilibria, in which the assumption is that everyone is fully aware of the strategies of the other players, but the aim is for all players to maximize their own gain.

Our results concern the safe game of competitive diffusion played on trees. We give an optimal safe strategy for full q -ary trees, and give asymptotically optimal safe strategies for spiders. For spiders consisting of a number of paths of equal length joined at a common vertex, we show that the safety value equals the gain of the disadvantaged player in a Nash equilibrium for competitive diffusion. In other words, we cannot improve on the safe gain by assuming the fully open and self-interested game rather than the adversarial setting.

Finally, we use results for special types of trees in order to develop a heuristic algorithm that can be applied to any tree. We show that the algorithm gives optimal results when applied to certain subclasses of trees. We also test the algorithm on randomly generated trees and show that the safe strategies found by the algorithm perform close to optimally.

1.1. Related Work

The first studies on the spread of influence through social networks assumed a passive model. The goal was to predict how information diffuses through a network starting from a given set of vertices. If the information reaches a vertex, this vertex is said to be activated. There are two main types of diffusion models: threshold models and cascade models. The difference in these models is in how vertices become activated.

In threshold models, vertices become activated once a variable associated with the neighborhood of a vertex surpasses a certain threshold. The most commonly used is the linear threshold model (see [7] and [9]). In this model, each vertex v has a threshold θ_v , and a vertex v is influenced by each of its neighbors, w , by a weight $b_{v,w}$. A vertex becomes activated once the sum of the weights of its activated neighbours exceeds θ_v .

In cascade models, as a vertex becomes activated, it activates each of its neighbors with a given probability. The most well-known is the independent cascade model (see [5] and [9]). In this model, we also start with an initial set of activated vertices. Here, each edge vw is assigned a probability $p_{v,w}$. If vertex v becomes activated, its neighbor w will become activated in the next round with probability $p_{v,w}$. The spread of influence in competitive diffusion can be seen as a cascade model in which the activation probability equals 1.

The optimization problem studied in these diffusion models is to choose the set of starting vertices so that the expected diffusion is maximized (see [9], [3], and [16]). In other words, the goal is to identify a set of initial influenced users who will bring a greater overall influence throughout the network. A related approach is through Voronoi games on graphs (see [22] and [2]). Here the players choose a set of vertices, and all other vertices are assigned to the starting vertex that is closest to it.

Competitive diffusion, as proposed in [1], is the first game-theoretic model in which the players are considered to be outside the social networks. Players choose initial users to influence and their goal is to reach the most users. In [1] (see also erratum [21]), the authors discuss the relationship between the diameter of the graph and the existence of pure Nash equilibria. A pure Nash equilibrium is a strategy that corresponds to a set of initial vertices, whereas a mixed strategy represents a probabilistic approach wherein starting vertices are chosen with a certain probability. In [20], the existence of a pure Nash equilibrium for competitive diffusion on trees is shown, whereas in [17], results on pure Nash equilibria are given for several classes of graphs. Moreover, [19] considers the competitive diffusion on a recently proposed model for online social networks and discusses the existence of Nash equilibria. The safe game for competitive diffusion was introduced in [4], and some results for paths were given.

Generalizations of competitive diffusion were proposed in [6] and [23]. In [6], the agents choose an allocation of budgeted seeds over the vertices, and the diffusion process is stochastic. In [23], the agents choose an initial set of vertices and the diffusion is a threshold model.

2. PRELIMINARIES

2.1. Competitive Diffusion Model

Let us start by recalling the model competitive diffusion from [1]. Let G be a graph with n vertices and suppose there are p players, P_1, \dots, P_p , each having a distinct assigned color (not white or grey). The strategy of each player is to choose a vertex in G as her/his starting vertex. Initially, all vertices are white. The game begins by coloring each of the starting vertices of the players and then proceeds by diffusing colors through G as follows: at each wave of diffusion, a vertex that has one or more neighbors with a certain color inherits that color while a vertex that has two neighbors with different colors turns grey. The diffusion finishes when all the vertices have either inherited a color, turned grey, or been forced to stay uncolored (white), due to being blocked off by grey vertices. In the end, the gain of each player is the number of vertices that has assumed his or her color. The winner of the game is the player that has the greatest gain. We note that if two or more players have the same starting vertex, then this vertex immediately turns grey.

Although the game can be played with any finite number of players, this article concentrates on the two-player version of the game. In the following, the two players will be called Player 1 (She) and Player 2 (He).

2.2. Mixed Strategies

Consider competitive diffusion on an undirected graph G with vertex set $V(G) = \{v_1, v_2, \dots, v_n\}$. We will denote the game matrix of Player 1 by A_G . Precisely, this is the matrix so that the entry $(A_G)_{ij}$ gives Player 1's gain if she chooses starting vertex v_i and Player 2 chooses starting vertex v_j . A mixed strategy for a player is a vector (x_1, x_2, \dots, x_n) so that $\sum_{i=1}^n x_i = 1$ and $x_i \geq 0$ for $i = 1, \dots, n$. It should be interpreted as a probabilistic strategy, where x_i is the probability the player chooses vertex v_i as the starting vertex. Accordingly, the expected gain of Player 1 when she plays the mixed strategy $X = (x_1, x_2, \dots, x_n)$ and Player 2 plays the mixed strategy $Y = (y_1, y_2, \dots, y_n)$ is

$$\text{Gain}(G, X, Y) = X A_G Y^T. \quad (2.1)$$

Let $\mathcal{S}_n = \{(z_1, z_2, \dots, z_n) \mid z_i \geq 0, 1 \leq i \leq n, \sum_{i=1}^n z_i = 1\}$ be the strategy set of the players. We will use the special notation $Z(v_k)$ for a mixed strategy equivalent to a pure strategy, i.e., when a player chooses vertex $v_k \in V(G)$ with probability 1 and the other vertices with probability 0. Precisely, $Z(v_k) = (z_1, z_2, \dots, z_n)$ with

$$z_i = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

2.3. Safe Game and Notations

The safe game for competitive diffusion is the zero-sum game in which the game matrix is A_G , which is the game matrix for Player 1 in competitive diffusion. The *safety value* for Player 1 is

$$\text{value}(A_G) = \min_{Y \in \mathcal{S}_n} \max_{X \in \mathcal{S}_n} X A_G Y^T = \max_{X \in \mathcal{S}_n} \min_{Y \in \mathcal{S}_n} X A_G Y^T.$$

Moreover, if $\text{Gain}(G, X^*, Y^*) = \text{value}(A_G)$, then X^* is called the *maxmin strategy* for Player 1, and Y^* is called the *minmax strategy* for Player 2. The existence of strategies X^* and Y^* and the uniqueness of $\text{value}A_G$ is guaranteed by Nash’s Existence theorem [15]. Indeed, X^* and Y^* form a Nash equilibrium in the safe game.

Any (mixed) strategy for Player 1 in the safe game will be referred to as a safe strategy. Correspondingly, any mixed strategy for Player 2 in the safe game will be called an opposing strategy. In this article, all strategies are assumed to refer to the safe game, unless stated otherwise.

The *guaranteed gain* of Player 1 with the safe strategy X , $\text{GGain}(G, X)$, is the minimum gain that Player 1 could receive with the strategy X , i.e.,

$$\text{GGain}(G, X) = \min_{Y \in \mathcal{S}_n} X A_G Y^T = \min_{Y \in \mathcal{S}_n} \text{Gain}(G, X, Y) = \min_{v_i \in V(G)} \text{Gain}(G, X, Z(v_i)). \tag{2.3}$$

The last step follows because any mixed strategy is a weighted average of pure strategies, thus, the minimum is always achieved by a pure strategy. A similar argument explains the last step in Equation (2.4), to follow.

The *maximal gain* of Player 1 against the opposing strategy Y of Player 2 is the maximum gain that Player 1 could receive when Player 2 chooses the strategy Y , i.e.,

$$\text{MGain}(G, Y) = \max_{X \in \mathcal{S}_n} X A_G Y^T = \min_{X \in \mathcal{S}_n} \text{Gain}(G, X, Y) \max_{v_i \in V(G)} \text{Gain}(G, Z(v_i), Y). \tag{2.4}$$

Note also that the guaranteed gain of Player 1 with any pure strategy $Z(v_k)$ equals zero, because Player 2 can counter by playing the same strategy $Z(v_k)$, reducing the gain of Player 1 to zero. Thus, any optimal safe strategy will be mixed.

The guaranteed gain with any safe strategy X for Player 1 is a lower bound on the safety value whereas the maximum gain of Player 1 against any opposing strategy Y of Player 2 is an upper bound on the safety value. Mathematically, we have

$$\text{GGain}(G, X) \leq \text{value}(A_G) \leq \text{MGain}(G, Y). \tag{2.5}$$

Thus, any pair of strategies for Player 1 and Player 2 give a lower and upper bound on the safe gain $value(A_G)$. In the following sections, we find strategies so that these bounds are tight or asymptotically tight.

2.4. Trees: Weights and Centroid

In this work, we study the safe game for competitive diffusion on trees. Here we introduce some facts about trees that are relevant to our analysis.

There exists more than one notion of center in a graph. We use the “branch weight” notion of centroid from [14]. A *branch* of a tree T at a vertex v is a maximal subtree of T , which has v as a leaf. Correspondingly, the *weight* of the vertex v , $w(v)$, is the maximum number of edges in any branch of v . The *centroid* of T , denoted $C(T)$, is the set of vertices that have the minimum weight in T . Unless stated otherwise, we will use n to denote the *order* (number of vertices) of the tree.

In a tree, it is known that the centroid is either a single vertex or two adjacent vertices [10]. Moreover, a tree that has only one vertex as centroid is called a *centroidal tree* and a tree that has two vertices as centroid is called a *bicentroidal tree*. We also have the following condition for a vertex to be in the centroid of a tree.

Theorem 2.1. (from [8]) *Let T be a tree of order n , and let v be a vertex of T . Suppose there are k branches at v having n_1, n_2, \dots, n_k edges, respectively. Vertex v is a centroid vertex of T if and only if $n_i \leq \frac{n}{2}$ for $1 \leq i \leq k$.*

The following lemmas on the weights of vertices will be helpful in establishing the main results.

Lemma 2.2. *For any tree T of order n , if v is a vertex of T not part of the centroid $C(T)$, then its weight $w(v)$ is the number of edges in the branch at v in which $C(T)$ is located.*

Proof. By way of contradiction, let B be the branch at v in which the centroid is located, let B' be the branch at v' that achieves the maximum weight, and assume B and B' are distinct. Therefore, the edge sets of B and B' are disjoint, thus, $|E(B)| + |E(B')| \leq n - 1$.

Let c be a vertex in the centroid. So one of the branches at c contains v , therefore, it contains all edges in branches at v other than B . Because the weight of a vertex is the maximum number of edges in one of its branches, we have $w(c) \geq n - 1 - |E(B)|$.

On the other hand, because B is not the branch at v with the maximum number of edges, we have $w(v) = |E(B')| \leq n - 1 - |E(B)|$. Thus, $n - 1 - |E(B)| \leq w(c) \leq w(v) \leq n - 1 - |E(B)|$ because centroidal vertices have minimum weight in T . Hence, $w(c) = w(v)$, which is a contradiction because v is not a vertex in the centroid of T . \square

We will use the notation $\bar{w}(v) = n - w(v)$, where n is the order of T . Given a vertex v , if B is the branch at v that has the maximum number of edges, then $w(v) = |E(B)| = |V(B)| - 1$. Thus $\bar{w}(v) = n - |E(B)| = n - |V(B)| + 1$ is the number of vertices in the tree obtained from T by deleting all vertices of B except v .

Lemma 2.3. *For any tree T of order n , if v is a vertex of T not part of the centroid $C(T)$, then*

$$w(v) > \bar{w}(v). \tag{2.6}$$

Proof. If v is not a centroid vertex, at least one of its branches must have more than $\frac{n}{2}$ edges by Theorem 2.1. Since $w(v)$ is the number of edges in the largest branch at v , we must have $w(v) > \frac{n}{2}$. Thus $w(v) > n - w(v)$. \square

3. SPIDERS

We start the study of the two-player safe game of competitive diffusion by giving tight bounds on the safety value for the game on special cases of trees; spiders and complete trees. The corresponding safe strategies suggest a good safe strategy for a player on any tree.

A *spider* is a tree with one and only one vertex of degree exceeding 2. The vertex with degree exceeding 2 is called the *body* of the spider. Moreover, any branch at the body of the spider is none other than a nontrivial path and is called a *leg* of the spider; see [18].

Let us denote the m legs of a spider S by $\{s_1, s_2, \dots, s_m\}$ and their lengths, respectively, by $\{l(s_1), l(s_2), \dots, l(s_m)\}$. We will label a vertex v_i in S by an ordered pair (d, s) where d is the distance from the vertex v_i to the body of the spider and $s \in \{1, 2, \dots, m\}$ is the index of the leg the vertex belongs to. By convention, the body of the spider will be identified by the ordered pair $(0, 0)$. We suggest the following safe strategy for Player 1 on a spider with legs of equal lengths. The strategy has positive probabilities of choosing the body of the spider and the first k vertices of the legs.

Definition 3.1. Given a spider, S , with m legs, each having ℓ vertices, let the vertices of S be labeled $v_0, v_1, \dots, v_{m\ell}$, where v_0 is the body of S , and for $d \in \{1, 2, \dots, \ell\}$ and $s \in \{1, 2, \dots, m\}$, the vertex v_i where $i = d + (s - 1)\ell$ is the vertex labeled by the ordered pair (d, s) . For any $k \in \{0, 1, \dots, \ell\}$, define the strategy $C_S(k)$ to be the strategy $(z_0, z_1, z_2, \dots, z_{m\ell})$ as follows. Consider vertex v_i with label (d, s) . Then its probability

$$z_i = \begin{cases} 0 & \text{if } k < d \leq \ell, \\ \frac{1}{mk+1} & \text{if } 0 \leq d \leq k. \end{cases} \tag{3.1}$$

Considering the strategy $C_S(k)$ as a safe strategy for Player 1 and as an opposing strategy for Player 2 leads to the following bounds on the safety value.

Theorem 3.2. *In the two-player competitive diffusion on S with m legs, each having ℓ vertices, the safety value of Player 1 is between $\ell - \frac{\sqrt{\ell}}{\sqrt{m}} + \mathcal{O}(1)$ and ℓ (asymptotic as ℓ goes to infinity).*

Proof. Assume the vertices of the spider are labeled as in the statement of Definition 3.1, that is, if vertex v_j has label (d, s) , then $j = d + (s - 1)\ell$. As a lower bound, we have the guaranteed gain of Player 1 with the strategy $C_S(k)$. As stated in (2.3), the guaranteed gain is the minimum gain of Player 1 over all the possible starting vertices for Player 2. Because

of symmetry, we need to consider only the body of the spider and the vertices on one of the legs of S . When Player 2 chooses a vertex other than the body, then Player 1 can play the body and gain all vertices except possibly those in one leg. Thus, Player 1 can achieve a gain of at least $n - \ell > n/2$ vertices. If Player 2 chooses the body, then Player 1 can gain only the vertices in the leg that contains the starting vertex, so the gain of Player 1 is at most $\ell < n/2$. Therefore, the minimum gain for Player 1 occurs when Player 2 chooses the body. Thus, $\text{GGain}(S, C_S(k)) = \text{Gain}(S, C_S(k), Z(v_0))$.

The expected gain of Player 1 when Player 2 chooses the body is

$$\text{Gain}(S, C_S(k), Z(v_0)) = \frac{1}{mk + 1} \left(0 + m \sum_{\delta=1}^k \left(\ell - \left\lfloor \frac{\delta}{2} \right\rfloor \right) \right).$$

Here, the summation is over the gain obtained when Player 1 chooses a vertex at distance δ from the body. Evaluating the sum, we obtain

$$\text{Gain}(S, C_S(k), Z(v_0)) = \begin{cases} \frac{m}{mk + 1} \left(k\ell - \frac{k^2}{4} \right) & \text{if } k \text{ is even} \\ \frac{m}{mk + 1} \left(k\ell - \frac{k^2}{4} + \frac{1}{4} \right) & \text{if } k \text{ is odd.} \end{cases}$$

Maximizing $\text{GGain}(S, C_S(k))$ over k gives $k^* = \frac{2\sqrt{\ell}}{\sqrt{m}} + \mathcal{O}(1)$ as the optimal integer choice for k and $\text{GGain}(S, C_S(k^*)) = \ell - \frac{\sqrt{\ell}}{\sqrt{m}} + \mathcal{O}(1)$.

For the upper bound, we have the maximum gain of Player 1 when Player 2 has the strategy $C_S(k)$ with $k = 0$. If $k = 0$, the strategy of Player 2 is simply to choose the body of the spider. In this case, the maximum gain Player 1 can obtain is ℓ , the number of vertices in one leg. □

4. COMPLETE m -ARY TREES

A complete m -ary tree ($m \geq 2$) of height h , which we will denote by $T(m, h)$, is a rooted tree in which every internal vertex has exactly m children and all leaves have distance h to the root. The distance to the root of a vertex v is called the *depth* of v , and the set of all vertices with depth d is called *level* d . The number of vertices in $T(m, h)$ is $n = \frac{m^{h+1}-1}{m-1}$. Let us identify a vertex v_j by an ordered pair (d, e) , where d is the depth of v_j , and e is the position of the vertex in level d if the vertices in the level are numbered from left to right by $\{0, 1, 2, \dots, m^d - 1\}$. By convention, the root of the tree will be identified by the ordered pair $(0, 0)$.

In the following, we will use the notation $Z(d, e)$ to denote the pure strategy $Z(v_j)$, where v_j has label (d, e) .

We suggest the following safe strategy for Player 1 and opposing strategy for Player 2. The strategies have positive probabilities of choosing the root and the vertices in the first level of $T(m, h)$.

Definition 4.1. Let the strategy μ_1 be a mixed strategy (x_1, x_2, \dots, x_n) on $T(m, h)$ where $n = \frac{m^{h+1}-1}{m-1}$ and for all $i \in \{1, 2, \dots, n\}$,

$$x_i = \begin{cases} \alpha_1 = \frac{m^h - 1}{m^{h+2} - m^{h+1} + m^h - 1} & \text{if } v_i \text{ is the root,} \\ \beta_1 = \frac{(m-1)m^h}{m^{h+2} - m^{h+1} + m^h - 1} & \text{if } v_i \text{ has depth 1,} \\ 0 & \text{if } v_i \text{ has depth } d > 1. \end{cases}$$

The probabilities α_1 and β_1 in the strategy μ_1 were obtained by solving

$$\begin{aligned} \text{Gain}(T(m, h), \mu_1, Z(0, 0)) &= \text{Gain}(T(m, h), \mu_1, Z(1, 1)) \\ \Leftrightarrow \alpha_1 \cdot 0 + m\beta_1 \cdot \left(\frac{m^h - 1}{m - 1}\right) &= \alpha_1 \cdot m^h + \beta_1 \cdot 0 + (m - 1)\beta_1 \cdot \left(\frac{m^h - 1}{m - 1}\right), \end{aligned} \quad (4.1)$$

subject to the condition that $\alpha_1 + m\beta_1 = 1$.

Definition 4.2. Let the strategy μ_2 be a mixed strategy (y_1, y_2, \dots, y_n) on $T(m, h)$ where $n = \frac{m^{h+1}-1}{m-1}$ and for all $i \in \{1, 2, \dots, n\}$,

$$y_i = \begin{cases} \alpha_2 = \frac{(m-1)(m^{h+1} - m^h + 1)}{m^{h+2} - m^{h+1} + m^h - 1} & \text{if } v_i \text{ is the root,} \\ \beta_2 = \frac{m^h - 1}{m^{h+2} - m^{h+1} + m^h - 1} & \text{if } v_i \text{ has depth 1} \\ 0 & \text{if } v_i \text{ has depth } d > 1. \end{cases}$$

The probabilities α_2 and β_2 in the strategy μ_2 were obtained by solving

$$\begin{aligned} \text{Gain}(T(m, h), Z(0, 0), \mu_2) &= \text{Gain}(T(m, h), Z(1, 1), \mu_2) \\ \Leftrightarrow \alpha_2 \cdot 0 + m\beta_2 \cdot m^h &= \alpha_2 \cdot \left(\frac{m^h - 1}{m - 1}\right) + \beta_2 \cdot 0 + (m - 1)\beta_2 \left(\frac{m^h - 1}{m - 1}\right), \end{aligned} \quad (4.2)$$

subject to $\alpha_2 + m\beta_2 = 1$. In both these scenarios, it makes sense to equalize the expected gains because the players do not want to give an advantage to their opponent.

Considering the strategy μ_1 as a safe strategy for Player 1 on $T(m, h)$ and the strategy μ_2 as an opposing strategy for Player 2 on $T(m, h)$ leads to the following result.

Theorem 4.3. In the two-player game of competitive diffusion on $T(m, h)$, the safety value of Player 1 is

$$\frac{(n-1)((m-1)n+1)}{n(m^2-m+1)+m-1},$$

where $n = \frac{m^{h+1}-1}{m-1}$. Moreover, Player 1 achieves the greatest gain with the safe strategy μ_1 , and the best opposing strategy for Player 2 is strategy μ_2 .

Proof. As a lower bound, we have the guaranteed gain of Player 1 with the strategy μ_1 . It is determined similarly as for the spiders, by taking the minimum gain for Player 1 over all pure opposing strategies $Z(d, e)$. Due to symmetry, we need to consider only the root of the tree and one vertex of each level as possible starting vertices for Player 2. By (4.1) we have that

$$\begin{aligned} \text{Gain}(T(m, h), \mu_1, Z(0, 0)) &= \text{Gain}(T(m, h), \mu_1, Z(1, e)) \\ &= \alpha_1 \cdot 0 + m\beta_1 \left(\frac{m^h - 1}{m - 1} \right) \\ &= \frac{m^{h+1}(m^h - 1)}{m^{h+2} - m^{h+1} + m^h - 1}. \end{aligned} \quad (4.3)$$

If $2 \leq d \leq h$, so if Player 2 plays a vertex with depth at least 2, then Player 1 can play the body and win all vertices that are not in the subtree of the root in which the start vertex of Player 2 is located. Each subtree contains $\frac{m^h - 1}{m - 1}$ vertices, so Player 1 wins at least $n - \frac{m^h - 1}{m - 1} = m^h$ vertices. Moreover, m^h is greater than (4.3). Thus, $\text{GGain}(T(m, h), \mu_1) = \text{Gain}(T(m, h), \mu_1, Z(0, 0))$.

For the upper bound, we have the maximum gain of Player 1 when Player 2 chooses the strategy μ_2 . It is determined by taking the maximum gain over all pure strategies $Z(d, e)$ for Player 1. By (4.2) we have that

$$\begin{aligned} \text{Gain}(T(m, h), Z(0, 0), \mu_2) &= \text{Gain}(T(m, h), Z(1, e), \mu_2) \\ &= \alpha_2 \cdot 0 + m\beta_2 \cdot m^h \\ &= \frac{m^{h+1}(m^h - 1)}{m^{h+2} - m^{h+1} + m^h - 1}. \end{aligned}$$

Furthermore, for $2 \leq d \leq h$,

$$\text{Gain}(T(m, h), Z(d, e), \mu_2) \leq (\alpha_2 + \beta_2) \left(\frac{m^{h-1} - 1}{m - 1} \right) + (m - 1)\beta_2 \left(\frac{m^h - 1}{m - 1} \right).$$

Substituting the expressions for α_2 and β_2 , we can show that this gain is smaller than $\text{Gain}(T(m, h), Z(0, 0), \mu_2)$. Thus, $\text{MGain}(T(m, h), \mu_2) = \text{Gain}(T(m, h), Z(0, 0), \mu_2)$. Finally, we have

$$\text{GGain}(T(m, h), \mu_1) = \text{MGain}(T(m, h), \mu_2) = \frac{(n - 1)((m - 1)n + 1)}{n(m^2 - m + 1) + m - 1}.$$

The last equality follows by making the substitution $n = \frac{m^{h+1} - 1}{m - 1}$ in the common expression for $\text{GGain}(T(m, h), \mu_1)$ and $\text{MGain}(T(m, h), \mu_2)$. \square

5. AN ALGORITHM TO FIND SAFE STRATEGIES FOR TREES IN GENERAL

In this section we exploit our earlier results to develop a heuristic algorithm in order to find a good safe strategy for any tree. We assume our tree to be centroidal. Such a tree has the centroid as its root, and $n - 1$ vertices divided among a number of branches. For bicentroidal trees, we can adopt a similar approach by considering one of the two vertices of the centroid to be the root.

The branches extending from the centroid can have different configurations. In spiders, all the branches at the centroid are nontrivial paths. We showed that a safe strategy that chooses positive probability vertices on the branches up to a certain distance has a guaranteed gain near the safety value. However, the branches in a complete tree are more clustered. We showed that a safe strategy that has a guaranteed gain equal to the safety value chooses only the root and the first vertex of each branch. This suggests considering different types of branches at the centroid and defining, accordingly, a distribution of probabilities on the vertices in the branch.

5.1. Branches at the Centroid

We distinguish three different types of branches at the centroid.

Definition 5.1. *A thick branch at the centroid is a branch for which we have*

$$w_2 \geq n - w_1 + \frac{w_1^2}{n},$$

where w_2 is the second-lowest weight in the branch and w_1 is the lowest weight in the branch.

A medium branch at the centroid is a branch for which we have

$$w_2 < n - w_1 + \frac{w_1^2}{n} \text{ and } w_3 \geq n - w_2 + \frac{w_2^2 + (w_2 - w_1)^2}{n + (w_2 - w_1)},$$

where w_3 is the third-lowest weight in the branch, w_2 is the second-lowest weight in the branch, and w_1 is the lowest weight in the branch.

A thin branch at the centroid is a branch for which we have

$$w_2 < n - w_1 + \frac{w_1^2}{n} \text{ and } w_3 < n - w_2 + \frac{w_2^2 + (w_2 - w_1)^2}{n + (w_2 - w_1)}, \tag{5.1}$$

where w_3 is the third-lowest weight in the branch, w_2 is the second-lowest weight in the branch and w_1 is the lowest weight in the branch.

By considering that the weight of a vertex is the number of edges in the branch in which lies the centroid (see Lemma 2.2), one can show that the vertex with the lowest weight in a branch is adjacent to the centroid and that the vertex with the second-lowest weight is adjacent to the first. The vertex with the next-lowest weight in the branch could be adjacent to either of these vertices. However, condition (5.1) assures us that the vertex with the third-lowest weight in a thin branch is adjacent to the vertex with the second-lowest weight. It would be impossible to have condition (5.1) and the vertices with second- and third-lowest weights on two different branches at the vertex with the lowest weight, since this would imply $n - 1 = w_1 + (n - w_2) + (n - w_3)$.

Following an approach similar to that involved in finding the strategies μ_1 and μ_2 , we give an algorithm that, given any centroidal tree T of size n , assigns a distribution of probabilities on the vertices of a branch that depends on the type of the branch. Let B_i be a branch of T and let u_i , t_i , and s_i be the vertices in B_i such that $w(u_i) \leq w(t_i) \leq w(s_i) \leq w(v_k)$ for any other vertex v_k in B_i . As explained in the previous paragraph, u_i is adjacent to the root, and t_i is adjacent to u_i . Vertex s_i could be adjacent to u_i or t_i , but in case of a

thin branch, s_i is adjacent to t_i . The algorithm proposed here will assign a probability α to the root (centroid) and, in each branch B_i , assigns probabilities β_i , γ_i , and δ_i to the vertices u_i , t_i and s_i , respectively, and probability zero to all other vertices in B_i .

The probabilities β_i , γ_i , and δ_i are given below. The expressions are given in terms of the weights of vertices u_i , t_i , and s_i and of α , the probability assigned to the root vertex. The expression depends on whether the branch B_i is a thin, medium, or thick branch.

(i) If B_i is a thin branch,

$$\begin{aligned}\beta_i &= \left(\frac{\bar{w}(t_i)(w(u_i)\bar{w}(s_i) + (w(t_i) - w(s_i))(w(t_i) - w(u_i)))}{\bar{w}(s_i)\bar{w}(u_i)\bar{w}(t_i) + w(s_i)w(t_i)(w(s_i) - w(t_i))} \right) \alpha, \\ \gamma_i &= \left(\frac{w(t_i)}{\bar{w}(t_i)} \right) \beta_i, \text{ and} \\ \delta_i &= \left(\frac{w(s_i)}{\bar{w}(s_i)} \right) \gamma_i + \left(\frac{w(t_i) - w(u_i)}{\bar{w}(s_i)} \right) \alpha.\end{aligned}\tag{5.2}$$

(ii) If B_i is a medium branch,

$$\begin{aligned}\beta_i &= \left(\frac{w(u_i)}{\bar{w}(u_i)} \right) \alpha, \\ \gamma_i &= \left(\frac{w(t_i)}{\bar{w}(t_i)} \right) \beta_i, \text{ and} \\ \delta_i &= 0.\end{aligned}\tag{5.3}$$

(iii) If B_i is a thick branch,

$$\begin{aligned}\beta_i &= \left(\frac{w(u_i)}{\bar{w}(u_i)} \right) \alpha, \\ \gamma_i &= 0, \text{ and} \\ \delta_i &= 0.\end{aligned}\tag{5.4}$$

If B_i is a thin branch, the suggested probabilities are obtained by equalizing the expected gains of Player 1 when Player 2 chooses the centroid, the vertex u_i , the vertex t_i , and the vertex s_i , solving for β_i , γ_i , and δ_i . Note that all probabilities are expressed in terms of α , the probability Player 1 chooses as the centroid. This will allow for rescaling during the algorithm.

Similarly, the suggested probabilities if B_i is a medium branch were obtained by first setting δ_i to zero, then equalizing the expected gains of Player 1 when Player 2 chooses the centroid, the vertex u_i , and the vertex t_i . Finally, if B_i is a thick branch, we set γ_i and δ_i to zero and we equalize the expected gains of Player 1 when Player 2 chooses the centroid and the vertex u_i . The distribution of probabilities in the branches will be used in the suggested safe strategy for Player 1.

5.2. Centroidal Safe Strategy (CSS) Algorithm

A centroidal tree can have diverse proportions of thin, medium, and thick branches. Moreover, unlike spiders and complete trees, the number of vertices varies from branch to branch. Thus, some branches might have very few vertices compared to other branches. Therefore, it would be unreasonable to suggest a safe strategy for Player 1, which has positive probability of choosing vertices in every branch at the centroid. As such, we define an algorithm to suggest a safe strategy for Player 1. The algorithm starts by assigning positive probabilities to vertices on one branch and then disperses probabilities on other branches as long as it is beneficial to Player 1. To determine whether the assignment is beneficial, we define the criterion of a branch. This criterion will be used in the algorithm to order the branches and determine when the dispersion of probabilities on branches should stop.

Definition 5.2. For a branch B in a centroidal tree, we define the criterion of B , $Cr(B)$, as follows. If B has fewer than three vertices, $Cr(B) = 0$. Else, let u, t , and s be the vertices in B such that $w(u) \leq w(t) \leq w(s) \leq w(v)$ for any other vertex v in B . Then,

$$Cr(B) = \begin{cases} \bar{w}(u) & \text{if } B \text{ is a thick branch,} \\ \left(\frac{\bar{w}(t)}{n}\right) \bar{w}(u) + \left(\frac{w(t)}{n}\right) \bar{w}(t) & \text{if } B \text{ is a medium branch,} \\ \frac{w(t)\bar{w}(t)(n^2 - nw(s) - w(s)w(t) + w(t)^2 + 2w(s)w(u) - w(t)w(u))}{nw(t)\bar{w}(s) + w(u)w(t)(-n + w(s) + w(t)) + \bar{w}(t)w(u)^2} & \text{if } B \text{ is a thin branch.} \end{cases} \quad (5.5)$$

Algorithm 1 Centroidal Safe Strategy (CSS) Algorithm

INPUT: Centroidal tree, T , with d branches at the centroid.

STEP 1: Order the branches $\{B_1, B_2, \dots, B_d\}$ such that $Cr(B_i) \geq Cr(B_{i+1})$ for all $1 \leq i \leq d - 1$.

STEP 2: Build a sequence of safe strategies σ_i for Player 1 by considering each branch in order.

- (a) If $i = 0$, form σ_0 , a strategy where the centroid is chosen with probability α .
- (b) If $i > 0$, form σ_i , a safe strategy in which the centroid is chosen with probability α , the probabilities of choosing the vertices of the branches B_k , $1 \leq k < i$ are the same in terms of α as in the strategy σ_{i-1} and the probabilities of choosing the vertices in the branch B_i are as given by (5.2), if B_i is a thin branch, by (5.3), if B_i is a medium branch and by (5.4), if B_i is a thick branch.

STEP 3: Determine α by solving $\alpha + \sum_{j=1}^i (\beta_j + \gamma_j + \delta_j) = 1$ and calculate the expected gain of Player 1 with the strategy σ_i when Player 2 chooses the centroid,

$$\text{Gain}(T, \sigma_i, Z(c)) = \alpha \cdot 0 + \sum_{j=1}^i (\beta_j \cdot \bar{w}(u_j) + \gamma_j \cdot \bar{w}(t_j) + \delta_j \cdot \bar{w}(t_j)). \quad (5.6)$$

STEP 4:

- (a) If $i < d$ and $Cr(B_{i+1}) \geq \text{Gain}(T, \sigma_i, Z(c))$ return to STEP 2 with $i = i + 1$.

(b) If $i < d$ and $Cr(B_{i+1}) \leq Gain(T(n), \sigma_i, Z(c))$ or $i = d$, return the strategy σ_i and its expected gain, $Gain(T, \sigma_i, Z(c))$.

OUTPUT: Safe strategy for Player 1, σ_i , with expected gain $Gain(T, \sigma_i, Z(c))$.

A few explanations on the algorithm are needed. In **Step 4**, we return to **Step 2** to disperse probabilities on another branch if $Cr(B_{i+1}) \geq Gain(T, \sigma_i, Z(c))$. The criterion being greater than the current expected gain results in an increase of the expected gain of Player 1. Thus, the strategies σ_i give increased gain. This is shown in the following lemma.

Lemma 5.3. *In the CSS algorithm, if $Cr(B_{i+1}) \geq Gain(T, \sigma_i, Z(c))$, then*

$$Gain(T, \sigma_i, Z(c)) \leq Gain(T, \sigma_{i+1}, Z(c)) \leq Cr(B_{i+1}). \quad (5.7)$$

Proof. We refer here to the solved value of α in the strategy σ_i by $\alpha^{(i)}$, and let strategy σ_i be represented by the vector $\alpha^{(i)}(x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$. To form the strategy σ_{i+1} , probabilities of $\beta_{i+1}, \gamma_{i+1}$ and δ_{i+1} , were assigned to the vertices u_{i+1}, t_{i+1} and s_{i+1} of B_{i+1} . These probabilities were all defined in terms of α . For clarity, let $\beta_{i+1} = \beta_{i+1}^* \alpha$, $\gamma_{i+1} = \gamma_{i+1}^* \alpha$, and $\delta_{i+1} = \delta_{i+1}^* \alpha$, where the expressions for $\beta_{i+1}^*, \gamma_{i+1}^*$, and δ_{i+1}^* for the three types of branches can be found in (5.4), (5.3) and (5.2). Finally, to form the strategy, a new value for $\alpha, \alpha^{(i+1)}$ was computed to make sure the probabilities add to one.

Thus, we have the following relation between the probability vectors.

$$x_j^{(i+1)} = \begin{cases} x_j^{(i)} & \text{if } v_j \notin \{u_{i+1}, t_{i+1}, s_{i+1}\} \\ \beta_i^* \alpha^{(i+1)} & \text{if } v_j = u_{i+1}, \\ \gamma_i^* \alpha^{(i+1)} & \text{if } v_j = t_{i+1}, \\ \delta_i^* \alpha^{(i+1)} & \text{if } v_j = s_{i+1}. \end{cases}$$

and

$$\frac{1}{\alpha^{(i+1)}} = \sum_{j=1}^n x_j^{(i+1)} = \frac{1}{\alpha^{(i)}} + \beta_{i+1}^* + \gamma_{i+1}^* + \delta_{i+1}^*. \quad (5.8)$$

Let $v_k \in \{u_{i+1}, t_{i+1}, s_{i+1}\}$. Then we know that $x_k^{(i)} = 0$, since, initially, there were no positive probabilities on the vertices of the branch B_{i+1} . Thus we have

$$\begin{aligned} Gain(T, \sigma_{i+1}, Z(c)) &= \sum_{j=1}^n \alpha^{(i+1)} x_j^{(i+1)} \cdot Gain(T, Z(v_j), Z(c)) \\ &= \sum_{j=1}^n \alpha^{(i+1)} x_j^{(i)} Gain(T, Z(v_j), Z(c)) \\ &\quad + \beta_{i+1}^* \alpha^{(i+1)} Gain(T, Z(u_{i+1}), Z(c)) \\ &\quad + \gamma_{i+1}^* \alpha^{(i+1)} Gain(T, Z(t_{i+1}), Z(c)) \\ &\quad + \delta_{i+1}^* \alpha^{(i+1)} Gain(T, Z(s_{i+1}), Z(c)). \end{aligned}$$

We obtain

$$\begin{aligned} \text{Gain}(T, \sigma_{i+1}, Z(c)) &= \frac{\alpha^{(i+1)}}{\alpha^{(i)}} \text{Gain}(T, \sigma_i, Z(c)) \\ &\quad + \beta_{i+1} \text{Gain}(T, Z(u_{i+1}), Z(c)) \\ &\quad + \gamma_{i+1} \text{Gain}(T, Z(t_{i+1}), Z(c)) \\ &\quad + \delta_{i+1} \text{Gain}(T, Z(s_{i+1}), Z(c)) \end{aligned} \tag{5.9}$$

where $\frac{\alpha^{(i+1)}}{\alpha^{(i)}} + \beta_i + \gamma_i + \delta_i = 1$, by (5.8). Thus, $\text{Gain}(T, \sigma_{i+1}, Z(c))$ is the convex combination of the four different specific gains. By assumption of the lemma, we have that $Cr(B_{i+1}) \geq \text{Gain}(T, \sigma_i, Z(c))$. Moreover, β_{i+1} , γ_{i+1} , and δ_{i+1} are defined so that the three last terms of the sum are at least as large as $Cr(B_{i+1})$. To verify this fact, we need to consider each type of branch separately. We give the explicit argument here only for a thick branch. The proof for the other types can be equally obtained by straightforward substitution of the appropriate values for β_{i+1} , γ_{i+1} , and δ_{i+1} .

Suppose B_{i+1} is a thick branch. In this case, $\gamma_i = \delta_i = 0$. Then $Cr(B_{i+1}) = \bar{w}(u_{i+1})$ (see Definition 5.2). Recall from Lemma 2.2 that $w(u_{i+1})$ is the number of edges in the branch at u_{i+1} , in which lies the centroid. Thus, there are $\bar{w}(u_{i+1})$ vertices in the branch B_{i+1} and so

$$\text{Gain}(T, Z(u_{i+1}), Z(c)) = \bar{w}(u_{i+1}) = Cr(B_{i+1}) \geq \text{Gain}(T, \sigma_i, Z(c)).$$

By (5.9), $\text{Gain}(T, \sigma_{i+1}, Z(c))$ is a convex combination of $\text{Gain}(T, Z(u_{i+1}), Z(c))$ and $\text{Gain}(T, \sigma_i, Z(c))$. Thus the inequalities stated in the lemma follow. □

The lemma allows us to explain the ordering of the branches. After k loops in the algorithm, suppose we have two branches at the centroid, B_i and B_j , such that $Cr(B_i) \geq Cr(B_j) \geq \text{Gain}(T, \sigma_k, Z(c))$. By Lemma 5.3, we know that adding positive probabilities on either of the branches B_i and B_j will increase the expected gain of Player 1. Moreover, the resulting expected gains will not surpass $Cr(B_i)$ and $Cr(B_j)$ respectively. If we start by adding the probabilities on the branch B_j , the branch B_i remains a candidate to increase the expected gain once more, since $Cr(B_i) \geq Cr(B_j)$. Let v be the resulting expected gain of Player 1 with the two branches added. However, if we start with branch B_i , the resulting expected gain might or might not be lower than $Cr(B_j)$. If it is, we can add the branch B_j to get the expected gain v . If it is not, then the resulting expected gain with only the branch B_i is greater than v . Thus, it is always advantageous to include the branch with the largest criterion first. For this reason, we order the branches in decreasing order of criterion in the algorithm.

Finally, we must show that the minimum expected gain of Player 1 with the strategy σ_i is obtained when Player 2 chooses the centroid.

Theorem 5.4. *Let T be a centroidal tree of size n with d branches at the centroid. Suppose we apply the safe strategy algorithm to T and we get the mixed strategy σ_k of Player 1 as output. Then,*

$$\text{GGain}(T, \sigma_k) = \text{Gain}(T, \sigma_k, Z(c))$$

where c is the centroid of T .

Proof. Suppose T has k_1 thick branches, $\{i_1^{\text{Thk}}, i_2^{\text{Thk}}, \dots, i_{k_1}^{\text{Thk}}\} \subseteq \{1, 2, 3, \dots, k\}$, k_2 medium branches, $\{i_1^{\text{Med}}, i_2^{\text{Med}}, \dots, i_{k_2}^{\text{Med}}\} \subseteq \{1, 2, 3, \dots, k\}$ and k_3 thin branches, $\{i_1^{\text{Thn}}, i_2^{\text{Thn}}, \dots, i_{k_3}^{\text{Thn}}\} \subseteq \{1, 2, 3, \dots, k\}$ from the set of branches $\{B_1, B_2, \dots, B_k\}$, where $k_1 + k_2 + k_3 = k$.

Proving this theorem consists of determining the expected gain of Player 1 over all the possible starting vertices for Player 2 and showing that the minimum occurs when Player 2 chooses the centroid. If Player 2 chooses to start with the centroid, the expected gain of Player 1 with the strategy σ_k is

$$\begin{aligned} \text{Gain}(T, \sigma_k, Z(c)) = & \alpha \cdot 0 + \sum_{j=i_1^{\text{Thk}}}^{i_{k_1}^{\text{Thk}}} \beta_j \cdot \bar{w}(u_j) + \sum_{j=i_1^{\text{Med}}}^{i_{k_2}^{\text{Med}}} (\beta_j \cdot \bar{w}(u_j) + \gamma_j \cdot \bar{w}(t_j)) \\ & + \sum_{j=i_1^{\text{Thn}}}^{i_{k_3}^{\text{Thn}}} (\beta_j \cdot \bar{w}(u_j) + \gamma_j \cdot \bar{w}(t_j) + \delta_j \cdot \bar{w}(t_j)). \end{aligned} \quad (5.10)$$

The gain for all other pure strategies of Player 2 can be determined from the definitions given earlier. The details of the proof are highly technical, and can be found in the Appendix. \square

6. EXPERIMENTAL ASSESSMENT OF THE CSS ALGORITHM

As a last section, we apply the CSS algorithm to some examples of centroidal trees in order to evaluate its guaranteed gain. We generated random centroidal trees with $n = 100$ and $n = 1000$ vertices. To evaluate the proximity of the guaranteed gain to the safety value, we calculate its difference to the maximum gain of Player 1 against a heuristic strategy for Player 2. This heuristic strategy chooses with positive probabilities, obtained experimentally, the centroid and some of the vertices at distances, 1 and 2 from the centroid. As explained in the first section, any opposing strategy for Player 2 gives an upper bound on the safety value. The smaller the gap between the safety value and any value obtained by a strategy of Player 2, the better the bound. The results show that in many cases the CSS algorithm renders a strategy that is close to optimal.

In our results, we give the gap between the opposing strategies as a proportion of the weight of the centroid. The reason for doing this, opposed to, for example, giving it as a proportion of the total number of vertices, n , is that we know for sure that the safety value of Player 1 is not greater than the weight of the centroid. This is the case because the maximal gain of Player 1 against the mixed strategy of Player 2 when he chooses the centroid with probability 1, is the weight of the centroid. This said, if we want to estimate, on a given centroidal tree, the difference between the payoff of the safe strategy from the CSS algorithm and the safety value, the percentage of the weight will give an estimate smaller than the weight of the centroid. Other wise if the difference is expressed as a proportion of n , we could have a tree with a large number of vertices but not necessarily a large centroid weight, for which the resulting estimation of difference could be bigger than the weight of the centroid, i.e., a difference greater than the safety value itself.

The trees were generated using MapleTM [11] and the computation of the CSS algorithm was carried out using MATLAB[®] [13]. The Maple function used for the generation of the trees is `RandomTree(n)` which has a randomized process as follows: “Starting with

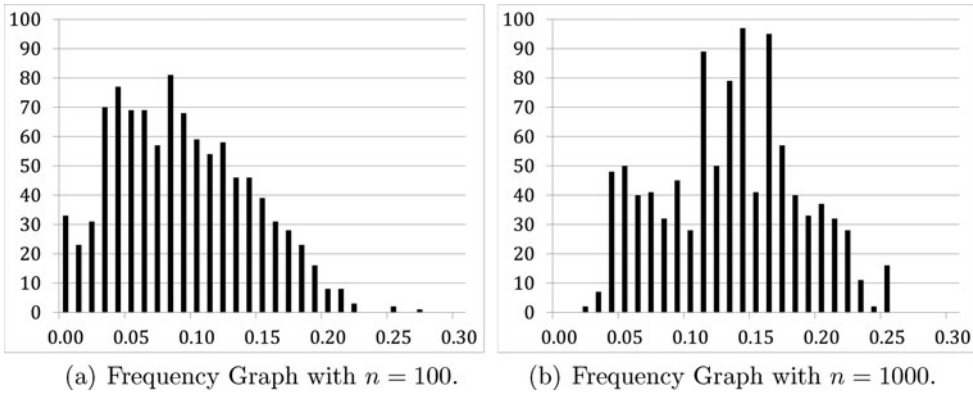


Figure 1 Frequency of the differences between the guaranteed gain of the CSS algorithm and the maximum gain of Player 1 as a proportion of the weight of the centroid in 1000 random centroidal tree examples with total number of vertices n .

the empty undirected graph T on n vertices, edges are chosen uniformly at random and inserted into T if they not create a cycle. This is repeated until T has $n - 1$ edges" [12]. Figure 1 shows the number of examples with a difference between the guaranteed gain obtained with the strategy given by the CSS algorithm, and the best upper bound. The difference is given as a proportion of the weight of the centroid. The columns represent the number of examples, out of a total of 1000, with a difference in the intervals $[0, 0]$, $(0, 0.01]$, $(0.01, 0.02]$, \dots , $(0.29, 0.30]$, respectively.

We see that the algorithm performs well and even ideally on a number of examples. A weaker performance was observed mostly for trees with large thin branches. This can be explained from our result for spiders, where all branches are thin. For spiders, the best strategy assigns positive probabilities to vertices at larger distance (about square root of the length) from the root. The CSS algorithm limits the number of vertices with positive probability in any branch to three. Thus, a trade-off of the versatility of the CSS algorithm is that the performance on thin branches is not optimal.

7. CONCLUSION AND FUTURE WORK

We have explored the safe game for competitive diffusion for trees. We obtained precise results for special classes of trees, namely spiders and complete trees. These results were then incorporated in the CSS algorithm, which can be applied to any tree. This algorithm was evaluated experimentally, and was shown to give good results.

However, performance of the CSS algorithm decreased with the presence of many thin branches. Generalizing the ideas presented here to include positive probabilities on more vertices on the branches would improve the algorithm. One might also consider slightly modifying the ordering of the branches or the distribution of the probabilities to compensate for the branches being considered isolated when the suggested distributions of probabilities were calculated.

We believe that the ideas put forward in this article can be extended beyond trees. Namely, the spread of influence can be seen as taking place in the form of a subtree of the graph. The general approach of assigning positive probabilities to only relatively few vertices close to the center of the graph, is likely to be of value here as well. Thus, it seems

plausible that the CSS algorithm might be modified to a more general setting. It might not be possible to obtain tight bounds for the safety value in the more general setting, but it should be possible to generate safe strategies that perform well in practice.

APPENDIX: PROOF OF THEOREM 5.4

Proof.

If Player 2 chooses to start with the centroid c , the expected gain of Player 1 with the strategy σ_k is as given earlier in (5.10). The labeling of the branches is as given after the statement of Theorem 5.4.

Let us now consider the cases when Player 2 chooses a vertex in a thin branch, B_r , $r \in \{i_1^{\text{Thn}}, i_2^{\text{Thn}}, \dots, i_{k_3}^{\text{Thn}}\}$.

- (i) If Player 2 chooses to start with the vertex u_r , the expected gain of Player 1 with the strategy σ_k is

$$\begin{aligned} \text{Gain}(T, \sigma_k, Z(u_r)) &= \alpha \cdot w(u_r) + \sum_{j=i_1^{\text{Thk}}}^{i_{k_1}^{\text{Thk}}} \beta_j \cdot \bar{w}(u_j) \\ &+ \sum_{j=i_1^{\text{Med}}}^{i_{k_2}^{\text{Med}}} (\beta_j \cdot \bar{w}(u_j) + \gamma_j \cdot \bar{w}(u_j)) \\ &+ \beta_r \cdot 0 + \gamma_r \cdot \bar{w}(t_r) + \delta_r \cdot \bar{w}(s_r) \\ &+ \sum_{j=i_1^{\text{Thn}}, j \neq r}^{i_{k_2}^{\text{Thn}}} (\beta_j \cdot \bar{w}(u_j) + \gamma_j \cdot \bar{w}(u_j) + \delta_j \cdot \bar{w}(t_j)). \quad (\text{A.1}) \end{aligned}$$

For all $j \in \{1, 2, \dots, k\}$, by definition we have that $w(u_j) < w(t_j)$, and thus

$$\bar{w}(u_j) > \bar{w}(t_j). \quad (\text{A.2})$$

Also, by the definition of β_r ,

$$\beta_r \cdot \bar{w}(u_r) + \delta_r \cdot \bar{w}(t_r) = \alpha \cdot w(u_r) + \delta_r \cdot \bar{w}(s_r). \quad (\text{A.3})$$

Using these, we can show that the expected gain (A.1) is greater than or equal to $\text{Gain}(T, \sigma_k, c)$ of (5.10).

- (ii) If Player 2 chooses vertex t_r , the expected gain of Player 1 is

$$\begin{aligned} \text{Gain}(T, \sigma_k, Z(t_r)) &= \alpha \cdot w(u_r) + \sum_{j=i_1^{\text{Thk}}}^{i_{k_1}^{\text{Thk}}} \beta_j \cdot w(u_r) \\ &+ \sum_{j=i_1^{\text{Med}}}^{i_{k_2}^{\text{Med}}} (\beta_j \cdot w(u_r) + \gamma_j \cdot \bar{w}(u_j)) \end{aligned}$$

$$\begin{aligned}
 & +\beta_r \cdot w(t_r) + \gamma_r \cdot 0 + \delta_r \cdot \bar{w}(s_r) \\
 & + \sum_{j=i_1^{Thn}, j \neq r}^{i_{k_3}^{Thn}} (\beta_j \cdot w(u_r) + \gamma_j \cdot \bar{w}(u_j) + \delta_j \cdot \bar{w}(u_j)). \tag{A.4}
 \end{aligned}$$

For all $j \in \{1, 2, \dots, k\}$, we have that

$$w(u_r) > \bar{w}(u_j). \tag{A.5}$$

If $j \neq r$, this follows since the branch at u_r in which the centroid is located includes the edges in the branch in which u_j is located. If $j = r$, we have the result by Lemma 2.3. Moreover,

$$\beta_r \cdot w(t_r) = \gamma_r \cdot \bar{w}(t_r)$$

by the definition of γ_r . Thus, the expected gain (A.4) is greater than or equal to $\text{Gain}(T, \sigma_k, u_r)$ of (5.10).

(iii) If Player 2 chooses vertex s_r , the expected gain of Player 1 is

$$\begin{aligned}
 \text{Gain}(T, \sigma_k, Z(s_r)) & = \alpha \cdot w(t_r) + \sum_{j=i_1^{Thk}}^{i_{k_1}^{Thk}} \beta_j \cdot w(u_r) \\
 & + \sum_{j=i_1^{Med}}^{i_{k_2}^{Med}} (\beta_j \cdot w(u_r) + \gamma_j \cdot w(u_r)) \\
 & + \beta_r \cdot w(t_r) + \gamma_r \cdot w(s_r) + \delta_r \cdot 0 \\
 & + \sum_{j=i_1^{Thn}, j \neq r}^{i_{k_3}^{Thn}} (\beta_j \cdot w(u_r) + \gamma_j \cdot w(u_r) + \delta_j \cdot \bar{w}(u_j)).
 \end{aligned}$$

This expected gain is greater than or equal to $\text{Gain}(T, \sigma_k, t_r)$ by (A.5) and since

$$\alpha \cdot w(u_r) + \delta_r \cdot \bar{w}(s_r) = \alpha \cdot w(t_r) + \gamma_r \cdot w(s_r)$$

by the definition of δ_r .

(iv) If Player 2 chooses to start with a vertex $v_j, v_j \notin \{u_r, t_r, s_r\}$, the payoff to Player 1 on all vertices not part of the branch B_r can only increase since Player 2's starting vertex is at a greater distance. Specifically, the payoff to Player 1 on the centroid is now at least $w(u_r)$. Moreover,

- If v_j is a descendant of u_r but not of t_r , and s_r then

$$\text{Gain}(T, Z(u_r), Z(v_j)) \geq w(u_r)$$

$$\text{Gain}(T, Z(t_r), Z(v_j)) \geq \bar{w}(t_r)$$

$$\text{Gain}(T, Z(s_r), Z(v_j)) \geq \bar{w}(s_r).$$

- If v_j is a descendant of u_r and t_r , but not of s_r then

$$\text{Gain}(T, Z(u_r), Z(v_j)) \geq w(u_r)$$

$$\text{Gain}(T, Z(t_r), Z(v_j)) \geq w(t_r)$$

$$\text{Gain}(T, Z(s_r), Z(v_j)) \geq \bar{w}(s_r).$$

- If v_j is a descendant of u_r , t_r , and s_r , then

$$\text{Gain}(T, Z(u_r), Z(v_j)) \geq w(u_r)$$

$$\text{Gain}(T, Z(t_r), Z(v_j)) \geq w(t_r)$$

$$\text{Gain}(T, Z(s_r), Z(v_j)) \geq w(s_r).$$

Since $w(v) > \bar{w}(v)$ for any vertex v other than the centroid by Lemma 2.3, in all cases we have

$$\begin{aligned} & \alpha \cdot \text{Gain}(T, Z(c), Z(v_j)) + \beta_r \cdot \text{Gain}(T, Z(u_r), Z(v_j)) \\ & + \gamma_r \cdot \text{Gain}(T, Z(t_r), Z(v_j)) + \delta_r \cdot \text{Gain}(T, Z(s_r), Z(v_j)) \\ & \geq \alpha \cdot w(u_r) + \beta_r \cdot w(u_r) + \gamma_r \cdot \bar{w}(t_r) + \delta_r \cdot \bar{w}(s_r) \\ & > \alpha \cdot w(u_r) + \gamma_r \cdot \bar{w}(t_r) + \delta_r \cdot \bar{w}(s_r) \\ & = \beta_r \cdot \bar{w}(u_r) + \gamma_r \cdot \bar{w}(t_r) + \delta_r \cdot \bar{w}(s_r), \end{aligned}$$

where the last equality follows from (A.3). Now,

$$\text{Gain}(T, Z(c), Z(c)) = 0,$$

$$\text{Gain}(T, Z(u_r), Z(c)) = \bar{w}(u_r),$$

$$\text{Gain}(T, Z(t_r), Z(c)) = \bar{w}(t_r) \text{ and}$$

$$\text{Gain}(T, Z(s_r), Z(c)) = \bar{w}(s_r),$$

Thus,

$$\begin{aligned} & \beta_r \cdot \bar{w}(u_r) + \gamma_r \cdot \bar{w}(t_r) + \delta_r \cdot \bar{w}(s_r) \\ & = \alpha \cdot \text{Gain}(T, Z(c), Z(c)) + \beta_r \cdot \text{Gain}(T, Z(u_r), Z(c)) \\ & + \gamma_r \cdot \text{Gain}(T, Z(t_r), Z(c)) + \delta_r \cdot \text{Gain}(T, Z(s_r), Z(c)). \end{aligned}$$

Therefore, the expected gain of Player 1 when Player 2 chooses the vertex v_j is greater than or equal to the expected gain of Player 1 when Player 2 chooses the centroid.

Similarly, we can show that the expected gain of Player 1 when Player 2 chooses to start with a vertex in a medium branch or thick branch is greater than the expected gain of Player 1 when Player 2 chooses to start with the centroid.

If Player 2 chooses a vertex in a branch B_i , $i > k$ instead of the centroid, Player 1's payoff on the vertices in the branches $\{B_1, B_2, \dots, B_k\}$ can only increase. In this case, strategy σ_k assigns no positive probabilities to any vertex in the branch. So, compared to the centroid, Player 2's starting vertex is at a greater distance from the vertices on which Player 1 has positive probability. Player 1's payoff on the centroid, being zero when Player 2 chooses the centroid, also increases. Thus, the expected gain of Player 1 is again greater.

In summation, the expected gain of Player 1 with the strategy σ_k is minimum when Player 2 chooses the centroid. Thus, $\text{Gain}(T, \sigma_k, Z(c))$ is the guaranteed gain of Player 1 with the strategy σ_k . \square

REFERENCES

- [1] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz. "A Note on Competitive Diffusion Through Social Networks." *Inf. Proc. Lett.* 110:6 (2010), 221–225.
- [2] S. Bandyapadhyay, A. Banik, S. Das, and H. Sarkar. "Voronoi Game on Graphs." In *WALCOM: Algorithms and Computation*, Vol. 7748 of Lecture Notes in Computer Science, pp. 77–88. Berlin, Heidelberg: Springer, 2013.
- [3] A. Borodin, Y. Filmus, and J. Oren. "Threshold Models for Competitive Influence in Social Networks." In *Internet and Network Economics*, Vol. 6484 of Lecture Notes in Computer Science, pp. 539–550. Berlin, Heidelberg: Springer, 2010.
- [4] D. Boudreau, J. Janssen, R. Nowakowski, and E. Roshanbin. Safe Strategies for Competitive Diffusion in Social Networks. Manuscript, 2012.
- [5] J. Goldenberg, B. Libai, and E. Muller. "Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth." *Marketing Letters* 12:3 (2001), 211–223.
- [6] S. Goyal and M. Kearns. "Competitive Contagion in Networks." In *Proceedings of the 44th Symposium on Theory of Computing, STOC '12*, pp. 759–774. New York, NY, USA: ACM, 2012.
- [7] M. Granovetter. "Threshold Models of Collective Behavior." *Am. J. of Soc.* 83:6 (1978), 1420–1443.
- [8] A. N. C. Kang and D. A. Ault. "Some Properties of a Centroid of a Free Tree." *Information Processing Letters* 4:1 (1975), 18–20.
- [9] D. Kempe, J. Kleinberg, and Éva Tardos. "Maximizing the Spread of Influence Through a Social Network." In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pp. 137–146, New York, NY, USA: ACM, 2003.
- [10] D. E. Knuth. *The Art of Computer Programming*. Reading, MA: Addison-Wesley, 1997.
- [11] Maplesoft, a division of Waterloo Maple Inc. *Maple 16*. Waterloo, Ontario: Author, 2012.
- [12] Maplesoft, a division of Waterloo Maple Inc. *Maple User Manual*. GraphTheory[RandomGraphs][RandomTree], (2005–2013). <http://www.maplesoft.com/support/help/Maple/view.aspx?path=GraphTheory/RandomGraphs/RandomTree>, Date accessed: January 23, 2014.
- [13] MathWorks, Inc. (The) *MATLAB and Statistics Toolbox Version 8.0.0.783 (R2012b)*. Natick, MA: Author, 2012.
- [14] S. L. Mitchell. "Another Characterization of the Centroid of a Tree." *Discrete Mathematics* 24:3 (1978), 277–280.
- [15] J. F. Nash. "Equilibrium Points in n-Person Games." *Proceedings of the National Academy of Sciences of the United States of America* 36:1 (1950), pp. 48–49.
- [16] C. Ning. "On the Approximability of Influence in Social Networks." *SIAM Journal on Discrete Mathematics* 23:3 (2009), 1400–1415.
- [17] E. Roshanbin. "The Competitive Diffusion Game in Classes of Graphs." In *Proceedings of 10th International Conference on Algorithmic Aspects of Information and Management (AAIM)*, Vancouver, BC, Canada: AAIM, 2014.
- [18] J.-L. Shang and C. Lin. "Spiders Are Status Unique in Trees." *Discrete Mathematics* 311:1011 (2011), 785–791.
- [19] L. Small and O. Mason. "Information Diffusion on the Iterated Local Transitivity Model of Online Social Networks." *Discrete Applied Mathematics* 161:1011 (2013), 1338–1344.
- [20] L. Small and O. Mason. "Nash Equilibria for Competitive Information Diffusion on Trees." *Information Processing Letters* 113:7 (2013), 217–219.

- [21] R. Takehara, M. Hachimori, and M. Shigeno. “A Comment on Pure-Strategy Nash Equilibria in Competitive Diffusion Games.” *Information Processing Letters* 112:3 (2012), 59–60.
- [22] S. Teramoto, E. D. Demaine, and R. Uehara. “The Voronoi Game on Graphs and Its Complexity.” *JGAA* 15:4 (2011), 485–501.
- [23] V. Tzoumas, C. Amanatidis, and E. Markakis. “A Game-Theoretic Analysis of a Competitive Diffusion Process over Social Networks.” In *Internet and Network Economics*, Vol. 7695 of Lecture Notes in Computer Science, pp. 1–14. Berlin, Heidelberg: Springer, 2012.