Competitive Diffusion on Weighted Graphs

Takehiro Ito[1,2], Yota Otachi[3], Toshiki Saitoh[2,4], Hisayuki Satoh[1], Akira Suzuki[1,2], Kei Uchizawa[5(✉)], Ryuhei Uehara[3], Katsuhisa Yamanaka[6], and Xiao Zhou[1]

[1] Tohoku University, Sendai, Japan
{takehiro,h.satoh,a.suzuki,zhou}@ecei.tohoku.ac.jp
[2] CREST, JST, Saitama, Japan
saitoh@eedept.kobe-u.ac.jp
[3] Japan Advanced Institute of Science and Technology, Nomi, Japan
{otachi,uehara}@jaist.ac.jp
[4] Kobe University, Kobe, Japan
[5] Yamagata University, Yonezawa, Japan
uchizawa@yz.yamagata-u.ac.jp
[6] Iwate University, Morioka, Japan
yamanaka@cis.iwate-u.ac.jp

Abstract. Consider an undirected and vertex-weighted graph modeling a social network, where the vertices represent individuals, the edges do connections among them, and weights do levels of importance of individuals. In the competitive diffusion game, each of a number of players chooses a vertex as a seed to propagate his/her idea which spreads along the edges in the graph. The objective of every player is to maximize the sum of weights of vertices infected by his/her idea. In this paper, we study a computational problem of asking whether a pure Nash equilibrium exists in a given graph, and present several negative and positive results with regard to graph classes. We first prove that the problem is W[1]-hard when parameterized by the number of players even for unweighted graphs. We also show that the problem is NP-hard even for series-parallel graphs with positive integer weights, and is NP-hard even for forests with arbitrary integer weights. Furthermore, we show that the problem for forests of paths with arbitrary weights is solvable in pseudo-polynomial time; and it is solvable in quadratic time if a given graph is unweighted. We also prove that the problem is solvable in polynomial time for chain graphs, cochain graphs, and threshold graphs with arbitrary integer weights.

1 Introduction

Ideas, innovations or trends spread by interactions between individuals. Social networks such as Facebook and Twitter facilitate their diffusion; an idea of an influential individual spreads along the connections over a network, and a small number of initial seeds can yield widespread infection. Since we can employ the so-called word-of-mouth effect as a tool for viral marketing, analysis of the

F. Dehne et al. (Eds.): WADS 2015, LNCS 9214, pp. 422–433, 2015.
DOI: 10.1007/978-3-319-21840-3_35

dynamics and process of the diffusion receive increasing attention in computer science. A number of papers focus on a task for a single company that wishes to advertise their product through a network; they investigate a problem of finding key individuals for maximizing the largest expected infection based on a given stochastic model of diffusion process [9,19,20,22]. Another active line of research stems from a task for multiple competing companies which try to advertise their products through a network, where the diffusion process is set in a game-theoretic formulation [1–7,10,15,16,23,24].

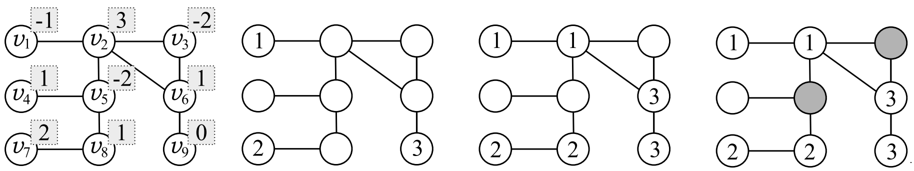
In this paper, we focus on the latter setting, and consider the one introduced by Alon *et al.* [1]. In their setting, a network is modeled by an unweighted graph, and each of a given number of competing companies chooses a vertex in the graph as a seed of their advertisement. Then their advertisements deterministically spread along the edges of a graph so that every infected vertex adopts its neighbors in a discrete time step. The objective of every player is to maximize the number of infected vertices. (The precise definition of the game is given in Section 2.) Alon *et al.* call the game *competitive diffusion game*, and show that there exists an unweighted graph of diameter three that does not admit a Nash equilibrium for two players. Following the paper [1], several results are known for the competitive diffusion game. Takehara *et al.* provided an unweighted graph G of diameter two that does not admit a Nash equilibrium for two players [24]. Small and Mason considered the case where a social network has a tree structure, and show that any tree admits a Nash equilibrium for two players [23]. More recently, Bulteau *et al.* consider certain graph classes including paths, cycles and grid graphs; in particular, they prove that there is no Nash equilibrium for three players on $m \times n$ grids with $\min\{m, n\} \geq 5$ [6].

We generalize the game to weighted graphs, where a weight on a vertex represents a level of importance of an individual; negative weights are admitted to express very demanding customers. We then focus on a problem Competitive Diffusion of deciding whether, given the number k, a graph G and weight function w, the competitive diffusion game on G with w for k players has a Nash equilibrium.

We establish solid complexity foundation of Competitive Diffusion with regard to graph classes. Since there are a number of theoretical models of social networks, and some of them are directly related to restricted graph classes (such as random trees with scale free properties [8]), our results give useful tools for obtaining algorithmic results on such models.

Our contributions are twofold. On the one hand, we provide the following three hardness results:

(i) Competitive Diffusion is W[1]-hard when parameterized by the number of players even for unweighted graphs;
(ii) Competitive Diffusion is NP-complete even for series-parallel graphs with positive integer weights;
(iii) Competitive Diffusion is NP-complete even for forests with arbitrary integer weights.



(a) G, w (b) Time 1 (c) Time 2 (d) Time 3

Fig. 1. Example of competitive diffusion with $k = 3$ players. (a) The graph G and weight w; numbers in the gray squares are weights. (b) p_1, p_2 and p_3 choose v_1, v_7 and v_9 in G, respectively; thus a strategy profile $\boldsymbol{s} = (v_1, v_7, v_9)$. (c) Each player dominates the neighbor. (d) The game ends; the two gray vertices are neutral. Consequently, $U_1(\boldsymbol{s}) = 2, U_2(\boldsymbol{s}) = 3$ and $U_3(\boldsymbol{s}) = 1$.

Very recently, Etesami and Basar studied unweighted version of the problem, and showed that Competitive Diffusion is a NP-complete problem [12], but their result does not imply ours. On the other hand, we obtain the following two algorithmic results.

(iv) For forests of paths, we prove that Competitive Diffusion is solvable in pseudo-polynomial time. In particular, we give a quadratic-time algorithm for forests of unweighted paths;

(v) For chain graphs, cochain graphs, and threshold graphs with arbitrary integer weights, we show that Competitive Diffusion is solvable in polynomial time.

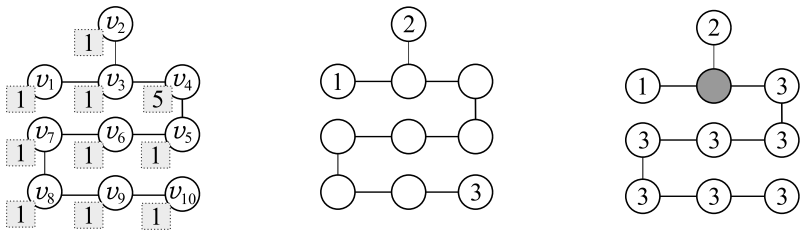
Note that, while four years past after Alon *et al.* introduced the competitive diffusion game, no nontrivial algorithm for the k-player game is known, even for unweighted trees with $k \geq 3$. Our research breaks this situation, and provides a new landscape of the computational aspect of the game.

The rest of the paper is organized as follows. In Section 2, we formally define the competitive diffusion game and the problem Competitive Diffusion. In Section 3, we present our hardness results for Competitive Diffusion. In Section 4, we give algorithms for forests of paths. In Section 5, we provide an algorithm for chain, cochain, and threshold graphs.

2 Preliminaries

We model a network as an undirected graph $G = (V, E)$, where the vertex set V represents individuals in the network, and the edge set E does the connections among them. The weight function $w : V \to \mathbb{Z}$ represents a level of importance of each individual. For a positive integer k, we define $[k] = \{1, 2, \ldots, k\}$, and call the k players $p_1, p_2, \ldots, p_k$.

The competitive diffusion game (k, G, w) proceeds as follows (see Fig. 1(a)–(d) for an explicit example). At time one, each player chooses a vertex in V; suppose a player p_i, $i \in [k]$, chooses a vertex $v \in V$. If any other player p_j,



(a) G, w (b) Time 1 (c) Time 7

Fig. 2. The vertex v_3 becomes neutral at time 2, and consequently, p_3 dominates v_4 at time 7

$i \neq j$, does not choose the vertex v, then p_i *dominates* v; and otherwise (that is, if there exists a player p_j, $i \neq j$, who chooses v), v becomes a *neutral* vertex. In the subsequent time steps, no player can dominate the neutral vertex. For each time t, $t \geq 2$, a vertex $v \in V$ is dominated by a player p_i at time t if (i) v is neither neutral nor dominated by any player by time $(t-1)$, and (ii) v has a neighbor dominated by p_i, but does not have a neighbor dominated by any player p_j, $i \neq j$. If v satisfies (i) and there are two or more players who dominate neighbors of v, then v becomes a neutral vertex at time t. The game ends when no player can dominate a vertex any more.

We note that the notion of a neutral vertex plays important role in the game; it sometimes gives critical effect on the result. (See Fig 2.) This contrasts to a similar game, called Voronoi game, where a player can dominate all the nearest vertices; if there is a vertex whose distances to seeds of two or more players tie, then they do not dominate but share the vertex [11,13,21,25].

Let $\boldsymbol{s} = (s^{(1)}, s^{(2)}, \dots, s^{(k)}) \in V^k$ be the vector of vertices which the players choose at the beginning of the game. We call $\boldsymbol{s}$ a *strategy profile*. For every $i \in [k]$, we define a *utility* $U_i(\boldsymbol{s})$ of p_i for $\boldsymbol{s}$ as the sum of the weights of the vertices which p_i dominates at the end. (See Fig. 1(d).)

For an index $i \in [k]$, we define $(\boldsymbol{s}_{-i}, v')$ as a strategy profile such that p_i chooses v' instead of $s^{(i)}$, but any other player p_j, $i \neq j$, chooses $s^{(j)}$: $(\boldsymbol{s}_{-i}, v') = (s^{(1)}, s^{(2)}, \dots, s^{(i-1)}, v', s^{(i+1)}, \dots, s^{(k)})$. For simplicity, we write $U_i(\boldsymbol{s}_{-i}, v')$ for $U_i((\boldsymbol{s}_{-i}, v'))$. Then, if $\boldsymbol{s}$ satisfies $U_i(\boldsymbol{s}_{-i}, v') \leq U_i(\boldsymbol{s})$ for every $i \in [k]$ and every $v' \in V$, we say that $\boldsymbol{s}$ is a *(pure) Nash equilibrium*. The strategy profile given in Fig. 1(b) is, in fact, a Nash equilibrium. We define Competitive Diffusion as the problem of deciding whether (k, G, w) has a Nash equilibrium.

3 Hardness Results on Competitive Diffusion

In this section, we observe computational complexity of Competitive Diffusion. Our first hardness result is the following theorem.

Theorem 1. Competitive Diffusion *is $W[1]$-hard even for unweighted graphs when parameterized by the number of players.*

To prove the theorem, we construct a reduction from a well-known $W[1]$-hard problem, Independent Set [14]. Given a graph $G = (V, E)$ and a positive integer k, Independent Set asks whether there exists an independent set I of size at least k, where a set I $(\subseteq V)$ is called an independent set if there is no pair of vertices $u, v \in I$ such that $(u, v) \in E$.

Below we provide the desired reduction and a proof overview.

Proof idea. We construct a graph $G' = (V', E')$ such that $G = (V, E)$ has an independent set I of size $|I| \geq k$ if and only if $(k + 3, G', w')$ has a pure Nash equilibrium, where $w' : V' \to \{1\}$.

Construction of G'

Let $n = |V|$, and d_v be the degree of v for every $v \in V$. The graph G' consists of two connected components $A = (V_A, E_A)$ and $B = (V_B, E_B)$.

We obtain the component A as follows. We construct a path of four vertices a_1, a_2, a_3, a_4; and make $2n$ vertices $a'_1, a'_2, \ldots, a'_n$ and $a''_1, a''_2, \ldots, a''_n$. Then we connect the terminal a_1 to $a'_1, a'_2, \ldots, a'_n$, and connect the terminal a_4 to $a''_1, a''_2, \ldots, a''_n$. We obtain the component B from the original graph G as follows. For every edge $e = (u, v) \in E$, we add a vertex b_e subdividing e. Then, for each $v \in V$, we introduce a set D_v of $n - d_v$ vertices, and connect v to every $u \in D_v$. Lastly we make a vertex b and λ vertices $b_1, b_2, \ldots, b_\lambda$, where λ is a sufficiently large number satisfying $\lambda = \Theta(n^3)$, and connect b to every $v \in V$, and connect b to $b_1, b_2, \ldots, b_\lambda$. Thus, we have $V' = V_A \cup V_B$ and $E' = E_A \cup E_B$.

Consider the game $(k+3, G', w')$. We can easily observe that any Nash equilibrium includes a strategy of a single player choosing the vertex b, since the strategy always give the maximum utility. Consequently, we can show that exactly two players can choose vertices other than the ones in the original graph G to hold a Nash equilibrium; otherwise, some player has extremely low utility (that is, below two) due to the player choosing b. In fact, we can show that any Nash equilibrium includes strategies of the two players choosing the vertex a_2 and a_3. Then the existence of a Nash equilibrium depends on whether there exists a strategy profile such that the other k players choose vertices composing an independent set: If the strategy profile of the other k players does not compose an independent set, then one of the k player obtains the utility less than $n + 1$; but the player can obtain the utility exactly $n + 1$ by changing its strategy to a_1 or a_4. □

For the cases where weights can be nonnegative or arbitrary integers, we can obtain the following stronger hardness results.

Theorem 2. Competitive Diffusion *is NP-complete even for series-parallel graphs with nonnegative integer weights.*

Theorem 3. Competitive Diffusion *is NP-complete even for forests of two components with integer weights.*

The proofs for Theorems 2 and 3 are similar to the one for Theorem 1, but we use other tricks by means of a neutral vertex together with positive and negative weights; we omit them due to the page limitation.

4 Algorithms for Forests of Paths

In the last section, we have shown that Competitive Diffusion is basically a computational hard problem. However, we can solve the problem for some particular graph classes. In Section 4.1, we give a pseudo-polynomial-time algorithm to solve Competitive Diffusion for forests of weighted paths; as its consequence, we show that the problem is solvable in polynomial time for forest of unweighted paths. In Section 4.2, we improve the running time of our algorithm to quadratic for the unweighted case.

4.1 Forests of Weighted Paths

Let F be a forest consisting of weighted m paths $P_1, P_2, \ldots, P_m$, and let W_j be the sum of the positive weights in a path P_j, $j \in [m]$. Then, we define $W = \max_{j \in [m]} W_j$ as the *upper bound on utility* for F, that is, any player can obtain at most W in F. In this subsection, we prove the following theorem.

Theorem 4. *Let F be a forest of weighted paths. Let n and W be the number of vertices in F and the upper bound on utility for F, respectively. Then, we can solve* Competitive Diffusion*, and find a Nash equilibrium, if any, in $O(Wn^9)$ time.*

We note that $W = O(n)$ if F is an unweighted graph. Therefore, by Theorem 4, Competitive Diffusion is solvable in $O(n^{10})$ time for an unweighted graph F; this running time will be improved to $O(n^2)$ in Section 4.2.

Idea and Definitions

Let F be a given forest consisting of weighted m paths $P_1, P_2, \ldots, P_m$. Let w be a given weight function; we sometimes denote by w_j the weight function restricted to the path P_j, $j \in [m]$. Suppose that, for an integer k, there exists a strategy profile $\boldsymbol{s}$ for the game (k, F, w) that is a Nash equilibrium. Then, the strategy profile restricted to each path P_j, $j \in [m]$, forms a Nash equilibrium for (k_j, P_j, w_j), where k_j is the number of players who chose vertices in P_j. However, the other direction does not always hold: A Nash equilibrium $\boldsymbol{s}_j$ for (k_j, P_j, w_j) is not always extended to a Nash equilibrium for the whole forest F, because some player may increase its utility by moving to another path in F. To capture such a situation, we classify a Nash equilibrium for a (single) path P_j more precisely.

Consider the game (κ_j, P_j, w_j) for an integer $\kappa_j \geq 0$. For a strategy profile $\boldsymbol{s}_j$ for (κ_j, P_j, w_j), we define $\mu_{P_j}(\boldsymbol{s}_j)$ as the minimum utility over all the κ_j players: $\mu_{P_j}(\boldsymbol{s}_j) = \min_{i \in [\kappa_j]} U_i(\boldsymbol{s}_j)$. In other words, any player in P_j obtains the utility at least $\mu_{P_j}(\boldsymbol{s}_j)$. For the case where $\kappa_j = 0$, we define $\boldsymbol{s}_j = \emptyset$ as the unique strategy profile for (κ_j, P_j, w_j); then, $\boldsymbol{s}_j$ is a Nash equilibrium and we define $\mu_{P_j}(\boldsymbol{s}_j) = +\infty$.

For a strategy profile $\boldsymbol{s}_j = \left(s_j^{(1)}, s_j^{(2)}, \ldots, s_j^{(\kappa_j)}\right)$ for (κ_j, P_j, w_j), we then define the "potential" of the maximum utility under $\boldsymbol{s}_j$ that can be expected to gain by an extra player other than the κ_j players. More formally, for a vertex v in P_j, we denote by $\boldsymbol{s}_j + v$ the strategy profile $\left(s_j^{(1)}, s_j^{(2)}, \ldots, s_j^{(\kappa_j)}, s_j^{(\kappa_j+1)}\right)$

for $(\kappa_j + 1, P_j, w_j)$ such that $s_j^{(\kappa_j+1)} = v$. Then, we define $\nu_{P_j}(\boldsymbol{s}_j) = \max_{v \in V(P_j)} U_{\kappa_j+1}(\boldsymbol{s}_j + v)$.

For two nonnegative integers κ_j and t, we say that P_j *admits* κ_j *players with a boundary* t if there exists a strategy profile $\boldsymbol{s}_j$ such that $\boldsymbol{s}_j$ is a Nash equilibrium for (κ_j, P_j, w_j) and $\nu_{P_j}(\boldsymbol{s}_j) \leq t \leq \mu_{P_j}(\boldsymbol{s}_j)$ holds. Then, the following lemma characterizes a Nash equilibrium of the game (k, F, w) in terms of the components of F; we omit the proof.

Lemma 1. *The game (k, F, w) has a Nash equilibrium if and only if there exist nonnegative integers $\kappa_1, \kappa_2, \ldots, \kappa_m$ and t such that $k = \sum_{j=1}^{m} \kappa_j$ and P_j admits κ_j players with the common boundary t for every $j \in [m]$.*

Algorithm

We first focus on a weighted single path.

Lemma 2. *Let P be a weighted path of n vertices, and t be a nonnegative integer. Then, one can find in $O(n^9)$ time the set $K \subseteq \{0, 1, \ldots, 2n\}$ of all the integers κ such that P admits κ players with boundary t.*

Based on Lemma 2, we can obtain the m sets $K_1, K_2, \ldots, K_m$, where $K_j \subseteq \{0, 1, \ldots, 2n\}$, $j \in [m]$, is the set of all the integers κ such that P admits κ players with boundary t. This can be done in $O(n^9)$ time, where n is the number of vertices in the whole forest F.

We now claim that, for a given integer t, it can be decided in $O(n^3)$ time whether there exist nonnegative integers $\kappa_1, \kappa_2, \ldots, \kappa_m$ such that $k = \sum_{j=1}^{m} \kappa_j$ and P_j admits κ_j players with the common boundary t for every $j \in [m]$; later we will apply this procedure to all possible values of t, $0 \leq t \leq W$. To show this, observe that finding desired m integers $\kappa_1, \kappa_2, \ldots, \kappa_m$ from the m sets $K_1, K_2, \ldots, K_m$ can be regarded as solving an instance of the multiple-choice knapsack problem [18]: The capacity c of the knapsack is equal to k; each integer κ' in K_j, $j \in [m]$, corresponds to an item with profit κ' and cost κ'; the items from the same set K_j form one class, from which at most one item can be packed into the knapsack. The multiple-choice knapsack problem can be solved in $O(cN)$ time [18], where N is the number of all items. Since $c = k$ and $N = O(mn)$, we can solve the corresponding instance in time $O(kmn) = O(n^3)$.

We finally apply the procedure above to all possible values of boundaries t. Since any player can obtain at most the upper bound W on utility for F, it suffices to consider $t \in [W]$. Therefore, our algorithm runs in $O(Wn^9)$ time in total.

4.2 Forests of Unweighted Paths

In this subsection, we improve the running time of our algorithm in Section 4.1 to quadratic when restricted to the unweighted case.

Theorem 5. *Let F be a forest of unweighted paths, and n be the number of vertices in F. Then, we can solve* Competitive Diffusion*, and find a Nash equilibrium, if any, in $O(n^2)$ time.*

In the rest of this subsection, we consider unweighted graphs, and thus define $w : V \to \{1\}$ for the vertex set V of a given forest. We assume that the number k of players is less than n; otherwise, a Nash equilibrium always exists. Note that, in this case, every player has utility at least one for any Nash equilibrium.

We first show that the set K_j of Lemma 2 can be obtained in $O(1)$ time, instead of $O(n^9)$ time, by characterizing Nash equilibriums for (κ, P, w) in terms of κ, t and n.

Lemma 3. *Let P be a single unweighted path of n vertices, and let κ and t be nonnegative and positive integers, respectively.*

(1) *P admits $\kappa = 0$ player with boundary t if and only if $n \le t$.*
(2) *P admits $\kappa = 1$ player with t if and only if $t \le n \le 2t + 1$.*
(3) *P admits $\kappa = 2$ players with t if and only if $2t \le n \le 2t + 2$.*
(4) *P admits $\kappa = 3$ players with t if and only if $t = 1$ and $n = 3, 4$ or 5.*
(5) *For any integer $\kappa \ge 4$, P admits κ players with t if and only if*

$$\begin{array}{ll} (\kappa+1)t - 1 \le n \le (2\kappa-4)t + \kappa & \textit{if } \kappa \textit{ is odd;} \\ \kappa t \le n \le (2\kappa-4)t + \kappa & \textit{if } \kappa \textit{ is even.} \end{array}$$

By Lemme 3, we can immediately obtain the number of players which P admits with a given boundary t:

Corollary 1. *Consider a fixed boundary t. If P is a path of n vertices, the numbers of players which P admits with a boundary t is given as follows.*

(1) *If $n \le t - 1$, the number is only 0.*
(2) *If $n = t$, the numbers are 0 and 1.*
(3) *If $t + 1 \le n \le 2t - 1$, the number is only 1.*
(4) *If $2t \le n \le 2t + 1$ and $n = 3$, the numbers are $1, 2$ and 3; and if $2t \le n \le 2t + 1$ and $n \ne 3$, the numbers are 1 and 2.*
(5) *If $n = 2t + 2$ and $n = 4$, the numbers are $2, 3$ and 4; and if $n = 2t + 2$ and $n \ne 4$, the number is only 2.*
(6) *If $2t + 3 \le n \le 4t - 1$, P has no desired Nash equilibrium.*
(7) *If $4t \le n$ and $5 \le n$, the numbers are integers κ such that*

$$\left\lceil \frac{n + 4t}{2t + 1} \right\rceil \le \kappa \le \max(k_{odd}, k_{even}),$$

where k_{odd} is the maximum odd integer satisfying $k_{odd} \le (n - t + 1)/t$, and k_{even} is the maximum even integer satisfying $k_{even} \le n/t$.

We use Corollary 1 to design our algorithm for forests of paths.

Without loss of generality, we assume that P_1 is a longest path among the m paths, and has n_1 vertices. For each t, $1 \le t \le n_1$, we repeat the following procedure: For every j, $1 \le j \le m$, we obtain, using Corollary 1, the minimum number $k_j^{\min}$ and the maximum number $k_j^{\max}$ of players which P_j admits with the boundary t. Corollary 1 implies that, for every j, $1 \le j \le m$, P_j admits κ

players with t for any κ between $k_j^{\min}$ and $k_j^{\max}$, and hence (k, F, w) has a Nash equilibrium with the common boundary t if and only if

$$\sum_{j=1}^{m} k_j^{\min} \leq k \leq \sum_{j=1}^{m} k_j^{\max}. \tag{1}$$

We thus complete the procedure by checking if the two inequalities in (1) both hold. Since Corollary 1 implies that we can obtain $k_j^{\min}$ and $k_j^{\max}$ in constant time for every j, the running time of the procedure above for single t is $O(m)$, and hence that of our entire algorithm is $O(n_1 m) = O(n^2)$, as desired.

5 Algorithms for Chain, Cochain, and Threshold Graphs

A bipartite graph $B = (X, Y; E)$ with $|X| = p$ and $|Y| = q$ is a *chain graph* if there is an ordering $(x_1, x_2, \ldots, x_p)$ on X such that $N(x_1) \subseteq N(x_2) \subseteq \cdots \subseteq N(x_p)$, where $N(u)$ denote a set of neighbors of a vertex u. If there is such an ordering on X, then there also exists an ordering $(y_1, y_2, \ldots, y_q)$ on Y such that $N(y_1) \subseteq N(y_2) \subseteq \cdots \subseteq N(y_q)$. We call such orderings *inclusion orderings*. A graph B' is a *cochain graph* if it can be obtained from a chain graph $B = (X, Y; E)$ by making the independent sets X and Y into cliques. A graph B'' is a *threshold graph* if it can be obtained from a chain graph $B = (X, Y; E)$ by making one of the independent sets X and Y into a clique. Observe that inclusion orderings on X and Y in B can be seen as *inclusion orderings* in B' and B'' if we use closed neighborhoods in cliques. Such inclusion orderings can be found in linear time [17]. Because the algorithm for chain graphs we will describe in this section depends only on its property of having inclusion orderings, we can apply the exactly same algorithm for cochain graphs and threshold graphs.

The following lemma follows directly from the definitions. Note that we denote $N[u] = N(u) \cup \{u\}$.

Lemma 4. *If $N(u) \subseteq N(v)$ or $N[u] \subseteq N[v]$ holds for $u = s^{(i)} \neq v = s^{(j)}$, then*

$$U_i(\boldsymbol{s}) = \begin{cases} 0 & \text{if there is } h \neq i \text{ such that } s^{(h)} = u, \\ w(u) & \text{otherwise.} \end{cases}$$

In what follows, let $B = (X, Y; E)$ be a chain graph with inclusion orderings $(x_1, \ldots, x_p)$ and $(y_1, \ldots, y_q)$ on X and Y, respectively. We define $\eta(\boldsymbol{s}, X) = \max(\{0\} \cup \{i \mid x_i \in V(\boldsymbol{s})\})$ and $\eta(\boldsymbol{s}, Y) = \max(\{0\} \cup \{i \mid y_i \in V(\boldsymbol{s})\})$.

Lemma 5. *Let $\boldsymbol{s}$ be a Nash equilibrium of B. If $s^{(i)} \notin \{x_{\eta(\boldsymbol{s},X)}, y_{\eta(\boldsymbol{s},Y)}\}$, then*

$$w(s^{(i)}) \geq \max\left\{w(u) \mid u \in \left(\{x_j \,|\, j \leq \eta(\boldsymbol{s}, X)\} \cup \{y_j \,|\, j \leq \eta(\boldsymbol{s}, Y)\}\right) \setminus V(\boldsymbol{s})\right\}. \tag{2}$$

Proof. Since $N(s^{(i)}) \subseteq N(x_{\eta(\boldsymbol{s},X)})$ or $N(s^{(i)}) \subseteq N(y_{\eta(\boldsymbol{s},Y)})$, it follows that $U_i(\boldsymbol{s}) \leq w(s^{(i)})$ by Lemma 4. Suppose for the contrary that there exists $u \in (\{x_j \mid j \leq \eta(\boldsymbol{s}, X)\} \cup \{y_j \mid j \leq \eta(\boldsymbol{s}, Y)\}) \setminus V(\boldsymbol{s})$ such that $w(s^{(i)}) < w(u)$.

Algorithm 1. Find a Nash equilibrium $\boldsymbol{s} \in V^k$ of a chain graph $B = (X, Y; E)$

Let $(x_1, \ldots, x_p)$ on X and $(y_1, \ldots, y_q)$ on Y be inclusion orderings.
// The following is for the case where $\eta(\boldsymbol{s}, X) \neq 0$.
for all guesses $(\eta(\boldsymbol{s}, X), \eta(\boldsymbol{s}, Y)) \in \{1, \ldots, p\} \times \{0, \ldots, q\}$ **do**
  $s^{(1)} := x_{\eta(\boldsymbol{s},X)}$. $s^{(2)} := y_{\eta(\boldsymbol{s},Y)}$ **if** $\eta(\boldsymbol{s}, Y) \neq 0$.
  $R := \{x_i \mid i < \eta(\boldsymbol{s}, X)\} \cup \{y_i \mid i < \eta(\boldsymbol{s}, Y)\}$.
  while there is a player i not assigned to a vertex **do**
    $v := \arg\max_{u \in R} w(u)$.
    if $w(v) \geq 0$ **then**
      $s^{(i)} := v$. $R := R \setminus \{v\}$.
    else
      $s^{(i)} := x_{\eta(\boldsymbol{s},X)}$.
    end if
  end while
  return $\boldsymbol{s}$ **if** it is a Nash equilibrium.
end for
return "no Nash equilibrium"

Now it holds that $N(u) \subseteq N(x_{\eta(\boldsymbol{s},X)})$ or $N(u) \subseteq N(y_{\eta(\boldsymbol{s},Y)})$. Thus, by Lemma 4, we have $U_i(\boldsymbol{s}_{-i}, u) = w(u) > w(s^{(i)}) \geq U_i(\boldsymbol{s})$. This contradicts the assumption that $\boldsymbol{s}$ is a Nash equilibrium. □

Thus, it suffices to check the strategy profiles satisfying Eq. (2) for our purpose.

Theorem 6. *Let G be a chain, cochain, or threshold graph of n vertices and m edges. Then, we can solve* Competitive Diffusion *for G, and find a Nash equilibrium, if any, in $O(n^4(m + n))$ time.*

Proof. We present an algorithm for chain graphs only. As previously described, we can apply the same algorithm for cochain and threshold graph.

We first guess $\eta(\boldsymbol{s}, X)$ and $\eta(\boldsymbol{s}, Y)$. Here we assume $\eta(\boldsymbol{s}, X) \neq 0$. The other case can be treated in the same way by swapping X and Y. We assign $x_{\eta(\boldsymbol{s},X)}$ to the first player. If $\eta(\boldsymbol{s}, Y) \neq 0$, then we assign $y_{\eta(\boldsymbol{s},Y)}$ to the second player. By Lemma 5, if $\boldsymbol{s}$ is a Nash equilibrium, then the other players have to select the heaviest vertices in $\{x_i \mid i < \eta(\boldsymbol{s}, X)\} \cup \{y_i \mid i < \eta(\boldsymbol{s}, Y)\}$. For each of the remaining players, we assign a vacant vertex with the maximum non-negative weight. If there is no such a vertex, we assign $x_{\eta(\boldsymbol{s},X)}$. We then test whether the strategy profile is a Nash equilibrium. See Algorithm 1.

Lemma 5 implies that if the algorithm assigns at most one player to $x_{\eta(\boldsymbol{s},X)}$, then the algorithm is correct. If two or more players are assigned to $x_{\eta(\boldsymbol{s},X)}$, then these players have utility 0. In such a case, there are not enough number of vertices of non-negative weights in $\{x_i \mid i < \eta(\boldsymbol{s}, X)\} \cup \{y_i \mid i < \eta(\boldsymbol{s}, Y)\}$. Thus every $\boldsymbol{s}$ with the guesses $\eta(\boldsymbol{s}, X)$ and $\eta(\boldsymbol{s}, Y)$ has a player with non-positive utility. If such a player, say p_i, has negative utility, then $\boldsymbol{s}$ is clearly not a Nash equilibrium. If p_i has utility 0, then it may improve its utility only if there is a vertex $v \in \{x_{\eta(\boldsymbol{s},X)+1}, \dots, x_p\} \cup \{y_{\eta(\boldsymbol{s},Y)+1}, \dots, y_q\}$ such that $U_i(\boldsymbol{s}_{-i}, v) > 0$. However, in this case, there is no Nash equilibrium with the guesses $\eta(\boldsymbol{s}, X)$ and $\eta(\boldsymbol{s}, Y)$. Therefore, the algorithm is correct.

We now analyze the running time of the algorithm. We have $O(n^2)$ options for guessing $x_{\eta(\boldsymbol{s},X)}$ and $y_{\eta(\boldsymbol{s},Y)}$. For each guess, the bottle-neck of the running time is to test whether the strategy profile is a Nash equilibrium or not. It takes $O(n^2(m+n))$ time as follows: we have $O(n^2)$ candidates of moves of players; for each candidate, we can compute the utility of the player moved by running a breadth-first search once in $O(m+n)$ time by adding a virtual root connecting to all the vertices occupied by the players. In total, the algorithm runs in $O(n^4(m+n))$ time. □

Acknowledgment. This work is partially supported by MEXT/JSPS KAKENHI, including the ELC project. (Grant Numbers: 24106004, 24106007 24106010, 24700130, 25330001, 25330003, 25730003, 26730001.)

References

1. Alon, N., Feldman, M., Procaccia, A.D., Tennenholtz, M.: A note on competitive diffusion through social networks. Information Processing Letters **110**(6), 221–225 (2010)
2. Apt, K.R., Markakis, E.: Social networks with competing products. Fundamenta Informaticae **129**(3), 225–250 (2014)
3. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 306–311. Springer, Heidelberg (2007)
4. Borodin, A., Braverman, M., Lucier, B., Oren, J.: Strategyproof mechanisms for competitive influence in networks. In: Proc. of the 22nd International Conference on World Wide Web, pp. 141–151 (2013)
5. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 539–550. Springer, Heidelberg (2010)
6. Bulteau, L., Froese, V., Talmon, N.: Multi-Player diffusion games on graph classes. In: Proc. of the 12th Annual Conference on Theory and Applications of Models of Computation (to appear)
7. Clark, A., Poovendran, R.: Maximizing influence in competitive environments: a game-theoretic approach. In: Proc. of the Second International Conference on Decision and Game Theory for Security, pp. 151–162
8. Cooper, C., Uehara, R.: Scale Free Properties of Random *k*-Trees. Mathematics in Computer Science **3**(4), 489–496 (2010)
9. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66 (2001)
10. Draief, M., Heidari, H., Kearns, M.: New models for competitive contagion. In: Proc. of the 28th AAAI Conference on Artificial Intelligence, pp. 637–644 (2014)
11. Dürr, C., Thang, N.K.: Nash equilibria in Voronoi games on graphs. In: Proc. of the 15th Annual European Symposium on Algorithms, pp. 17–28 (2007)

12. Etesami, S.R., Basar, T.: Complexity of equilibrium in diffusion games on social networks. 1403.3881
13. Feldmann, R., Mavronicolas, M., Monien, B.: Nash equilibria for voronoi games on transitive graphs. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 280–291. Springer, Heidelberg (2009)
14. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series (2006)
15. Goyal, S., Heidari, H., Kearns, M.: Competitive contagion in networks. Games and Economic Behavior (to appear)
16. He, X., Kempe, D.: Price of anarchy for the N-player competitive cascade game with submodular activation functions. In: Proc. of the 9th International Workshop on Internet and Network Economics, pp. 232–248 (2013)
17. Heggernes, P., Kratsch, D.: Linear-time certifying recognition algorithms and forbidden induced subgraphs. Nordic Journal of Computing **14**, 87–108 (2008)
18. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer-Verlag (2004)
19. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
20. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
21. Mavronicolas, M., Monien, B., Papadopoulou, V.G., Schoppmann, F.: Voronoi games on cycle graphs. In: Proc. of the 33rd International Symposium on Mathematical Foundations of Computer Science, pp. 503–514 (2008)
22. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70 (2002)
23. Small, L., Mason, O.: Nash Equilibria for competitive information diffusion on trees. Information Processing Letters **113**(7), 217–219 (2013)
24. Takehara, R., Hachimori, M., Shigeno, M.: A comment on pure-strategy Nash equilibria in competitive diffusion games. Information Processing Letters **112**(3), 59–60 (2012)
25. Teramoto, S., Demaine, E.D., Uehara, R.: The Voronoi game on graphs and its complexity. Journal of Graph Algorithms and Applications **15**(4), 485–501 (2011)