European Journal of Combinatorics 48 (2015) 88–99

Contents lists available at ScienceDirect

European Journal of Combinatorics

journal homepage: www.elsevier.com/locate/ejc

The maximum time of 2-neighbour bootstrap percolation: Algorithmic aspects[☆]

Fabrício Benevides[a], Victor Campos[b], Mitre C. Dourado[c], Rudini M. Sampaio[b], Ana Silva[a]

[a] Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, Brazil

[b] Departamento de Computação, Universidade Federal do Ceará, Fortaleza, Brazil

[c] Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Available online 5 March 2015

ABSTRACT

In 2-neighbourhood bootstrap percolation on a graph G, an infection spreads according to the following deterministic rule: infected vertices of G remain infected forever and in consecutive rounds healthy vertices with at least 2 already infected neighbours become infected. Percolation occurs if eventually every vertex is infected. In this paper, we are interested to calculate the maximal time $t(G)$ the process can take, in terms of the number of times the interval function is applied, to eventually infect the entire vertex set. We prove that the problem of deciding if $t(G) \geq k$ is NP-complete for: (a) fixed $k \geq 4$; (b) bipartite graphs and fixed $k \geq 7$; and (c) planar graphs. Moreover, we obtain linear and polynomial time algorithms for trees and chordal graphs, respectively.

1. Introduction

We consider a problem in which an infection spreads over the vertices of a connected simple graph G following a deterministic spreading rule in such a way that an infected vertex will remain infected forever. Given a set $S \subseteq V(G)$ of initially infected vertices, we build a sequence $S_0 = S, S_1, S_2, \ldots$ in which S_{i+1} is obtained from S_i using such a spreading rule.

☆ This research was supported by Capes (STIC-AmSud 040/2012) and CNPq (Universal 478744/2013-7).

E-mail addresses: fabricio@mat.ufc.br (F. Benevides), campos@lia.ufc.br (V. Campos), mitre@dcc.ufrj.br (M.C. Dourado), rudini@lia.ufc.br (R.M. Sampaio), anasilva@mat.ufc.br (A. Silva).

http://dx.doi.org/10.1016/j.ejc.2015.02.012
0195-6698/

Under r-neighbour bootstrap percolation on a graph G, the spreading rule is a threshold rule in which S_{i+1} is obtained from S_i by adding to it the vertices of G which have at least r neighbours in S_i. We say that a set S_0 percolates G (or that S_0 is a percolating set of G) if eventually every vertex of G becomes infected, that is, there exists a t such that $S_t = V(G)$. In that case, we define $t_r(S)$ as the minimum t such that $S_t = V(G)$. And define, the *percolation time of* G as $t_r(G) = \max\{t_r(S) : S \text{ percolates } G\}$. In this paper, we shall focus on the case where $r = 2$ and in such a case we omit the subscript of the functions $t_r(S)$ and $t_r(G)$.

Bootstrap percolation was introduced by Chalupa, Leath and Reich [14] as a model for certain interacting particle systems in physics. Since then it has found applications in clustering phenomena, sandpiles [23], and many other areas of statistical physics, as well as in neural networks [1] and computer science [19].

There are two broad classes of questions one can ask about bootstrap percolation. The first, and the most extensively studied, is what happens when the initial configuration S_0 is chosen randomly under some probability distribution? One would like to know how likely percolation is to occur, and if it does occur, how long it takes.

The answer to the first of these questions is now well understood for various graphs. An interesting case is the one of the lattice graph $[n]^d$, in which d is fixed and n tends to infinity, since the probability of percolation under the r-neighbour model displays a sharp threshold between no percolation with high probability and percolation with high probability. The existence of thresholds in the strong sense just described first appeared in papers by Holroyd, Balogh, Bollobás, Duminil-Copin and Morris [25,5,4]. Sharp thresholds have also been proved for the hypercube (Balogh and Bollobás [3], and Balogh, Bollobás and Morris [6]). There are also very recent results due to Bollobás, Holmgren, Smith and Uzzell [10], about the time percolation takes on the discrete torus $\mathbb{T}_n^d = (\mathbb{Z}/n\mathbb{Z})^d$ for a randomly chosen set S_0.

The second broad class of questions is the one of extremal questions. For example, what is the smallest or largest size of a percolating set with a given property? The size of the smallest percolating set in the d-dimensional grid $[n]^d$ was studied by Pete and a summary can be found in [7]. Morris [28] and Riedl [30] studied the maximum size of minimal percolating sets on the square grid $[n]^2$ and the hypercube $\{0, 1\}^d$, respectively, answering a question posed by Bollobás. However, the problem of finding the smallest percolating set is NP-hard even on subgraphs of the square grid [2] and it is APX-hard even for bipartite graphs with maximum degree four [17]. Moreover, it is hard [15] to approximate within a ratio $O(2^{\log^{1-\varepsilon} n})$, for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$.

Another type of question is: What is the minimum or maximum time that percolation can take, given that S_0 satisfies certain properties? Recently, Przykucki [29] determined the precise value of the maximum percolation time on the hypercube $2^{[n]}$ as a function of n, and Benevides and Przykucki [9,8] have similar results for the square grid $[n]^2$, also answering a question posed by Bollobás. In particular, they have a polynomial time dynamic programming algorithm to compute the maximum percolation time on rectangular grids [9].

In this paper, we investigate the computational complexity of $t(G)$, motivated by these recent results on the maximum percolation time. Here, we consider the decision version of the maximum time percolation problem, as stated below.

PERCOLATION TIME
Input: A graph G and an integer k.
Question: Is $t(G) \geq k$?

In Section 2, we prove that PERCOLATION TIME is NP-complete even when the input is restricted to certain cases. More precisely, we prove it is NP-complete for: general graphs even if $k \geq 4$ is fixed, that is, k is not part of the input; bipartite graphs and any fixed $k \geq 7$; and planar graphs and a given k, that is, k is part of the input. In Section 3, we provide polynomial time algorithms for general graphs when $k \leq 2$ and for chordal graphs, and a linear time algorithm for trees.

1.1. Related works and some notation

It is interesting to notice that infection problems appear in the literature under many different names and were studied by researches of various fields. A recent source on related topics is [16]. The

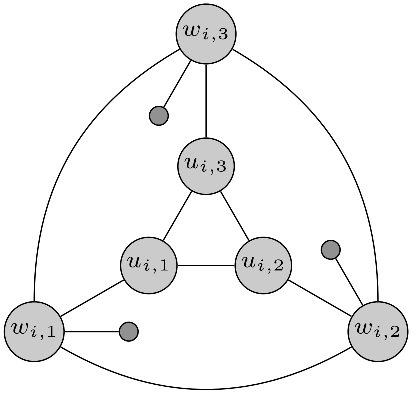

Fig. 1. Gadget for clause C_i.

particular case in which $r = 2$ in r-neighbourhood bootstrap percolation is also a particular case of a infection problem related to convexities in graph, which are also of our interest.

A finite *convexity space* [26,31] is a pair $(V, \mathcal{C})$ consisting of a finite ground set V and a set $\mathcal{C}$ of subsets of V satisfying $\emptyset, V \in \mathcal{C}$ and if $C_1, C_2 \in \mathcal{C}$, then $C_1 \cap C_2 \in \mathcal{C}$. The members of $\mathcal{C}$ are called $\mathcal{C}$*-convex sets* and the *convex hull* of a set S is the minimum convex set $H(S) \in \mathcal{C}$ containing S.

A convexity space $(V, \mathcal{C})$ is an *interval convexity* [11] if there is a so-called *interval function* $I : \binom{V}{2} \to 2^V$ such that a subset C of V belongs to $\mathcal{C}$ if and only if $I(\{x, y\}) \subseteq C$ for every two distinct elements x and y of C. With no risk of confusion, for any $S \subseteq V$, we also denote by $I(S)$ the union of S with $\cup_{x,y\in S} I(\{x, y\})$. In interval convexities, the convex hull of a set S can be computed by exhaustively applying the corresponding interval function until obtaining a convex set.

Given a vertex v of a graph G, $N(v)$ stands for the set of neighbours of v in G; for a nonnegative integer i, $N_i(v)$ denotes the set of vertices at distance i of v; and $\overline{N}(v)$ the set of vertices at distance at least two from v, that is, $\overline{N}(v) = V(G) \setminus (N(v) \cup \{u\})$.

The most studied graph convexities defined by interval functions are those in which $I(\{x, y\})$ is the union of paths between x and y with some particular property. Some common examples are the P_3-convexity [21], geodetic convexity [22] and monophonic convexity [20]. We observe that the spreading rule in 2-neighbours bootstrap percolation is equivalent to $S_{i+1} = I(S_i)$ where I is the interval function which defines the P_3-convexity: $I(S)$ contains S and every vertex belonging to some path of 3 vertices whose extreme vertices are in S. It will be convenient to denote S_i by $I^i(S)$, where $I^i(S)$ is obtained by applying i times the operation I.

For these reasons, sometimes we call a percolating set by *hull set*. Related to the geodetic convexity, there exists the *geodetic iteration number of a graph* [13,24], which is similar to the percolation time.

2. NP-complete cases

In the first case we make a reduction from the 3-SAT problem.

Theorem 2.1. PERCOLATION TIME *is NP-complete for any fixed* $k \geq 4$.

Proof. Given m clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ on variables $X = \{x_1, \ldots, x_n\}$ as an instance of 3-SAT, we denote the three literals of C_i by $\ell_{i,1}$, $\ell_{i,2}$ and $\ell_{i,3}$. We construct a graph G as follows.

Construction 1. For each clause C_i of $\mathcal{C}$, add to G a gadget as the one of Fig. 1. Then, for each pair of literals $\ell_{i,a}, \ell_{j,b}$ such that one is the negation of the other, add a vertex $y_{(i,a),(j,b)}$ adjacent to $w_{i,a}$ and $w_{j,b}$. Let Y be the set of all vertices created this way. Finally, add a vertex z adjacent to all vertices in Y and a pendant vertex z' adjacent to only z. Denote the sets $\{u_{i,1}, u_{i,2}, u_{i,3}\}$ and $\{w_{i,1}, w_{i,2}, w_{i,3}\}$ by U_i and W_i, respectively. Let $U = \cup_{1\leq i\leq m} U_i$, $W = \cup_{1\leq i\leq m} W_i$ and L be the set of vertices of degree one in G.

We first consider the case $k = 4$. We show that $\mathcal{C}$ is satisfiable if and only if G contains a hull set with percolation time at least 4.

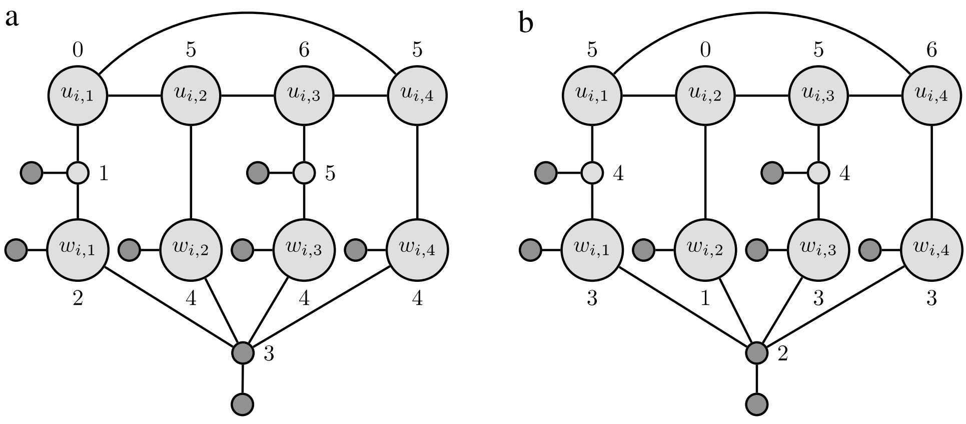

Fig. 2. Bipartite gadget for each clause C_i.

Suppose that $\mathcal{C}$ has a truth assignment. For each clause C_i, let k_i denote an integer in $\{1, 2, 3\}$ such that ℓ_{i,k_i} is true. Let $S' = \{u_{i,k_i} : 1 \leq i \leq m\}$ and $S = S' \cup L$. We obtain $I^1(S)$ from S by adding the vertices $\{w_{i,k_i} : 1 \leq i \leq m\}$; $I^2(S)$ from $I^1(S)$ by adding the remaining vertices in W; $I^3(S)$ from $I^2(S)$ by adding the vertices in Y together with the remaining vertices in U; and, $I^4(S)$ from $I^3(S)$ by adding the vertex z. Therefore, G has percolation time at least 4.

Now, suppose that $t(G) \geq 4$ and let S be any hull set of G with $t(S) \geq 4$. Note that $L \subseteq S$; also for any clause C_i, we have $U_i \cap S \neq \emptyset$ because $|N(u_{i,j}) - U_i| \leq 1$, for all i, j. This implies that $W \subseteq I^2(S)$, $U \cup Y \subseteq I^3(S)$ and $z \in I^4(S)$. Furthermore, if $Y \cap I^2(S) \neq \emptyset$ then $z \in I^3$ and $t(S) \leq 3$, a contradiction. Then $Y \cap I^2(S) = \emptyset$, which means that no pair $\{u_{i,a}, u_{i,b}\}$, where $\ell_{i,a}$ is the negation of $\ell_{j,b}$, is in S. This means that assigning true to each $\ell_{i,j}$ for which $u_{i,j} \in S$ gives us an assignment that satisfies $\mathcal{C}$.

For values $k > 4$, it suffices to subdivide the edge zz' into a path P of length $k - 4$, appending a new leaf vertex to each vertex in P. □

Theorem 2.2. PERCOLATION TIME *is NP-complete for bipartite graphs and any fixed* $k \geq 7$.

Proof. We describe a polynomial reduction from 4-SAT. Let $\mathcal{C}$ be an instance of 4-SAT with m clauses $C_1, \ldots, C_m$ on $X = \{x_1, \ldots, x_n\}$. We denote the four literals of a clause C_i of $\mathcal{C}$ by $\ell_{i,1}$, $\ell_{i,2}$, $\ell_{i,3}$ and $\ell_{i,4}$. We construct a graph G similar to the one obtained by Construction 1, using the gadget of Fig. 2 (a) instead of Fig. 1. Further, we add two vertices z'' and z''' and the edges zz'' and $z''z'''$. Also add a pendant vertex adjacent to only z'' and a pendant vertex adjacent to only z'''. Denote by A_i the vertices of the clause gadget associated to clause C_i, for $1 \leq i \leq m$. First consider the case $k = 7$. We show that $\mathcal{C}$ is satisfiable if and only if G contains a hull set with percolation time at least 7.

Suppose that $\mathcal{C}$ has a truth assignment. For each clause C_i, let k_i denote an integer in $\{1, 2, 3, 4\}$ such that ℓ_{i,k_i} is true. Let $S' = \{u_{i,k_i} : 1 \leq i \leq m\}$ and $S = S' \cup L$. Fig. 2(a) and (b) show the two possibilities for the percolation times of the vertices in the gadget clauses. From this observation, it is easy to see that the vertices in Y have percolation time either 4 or 5 (recall that $\mathcal{C}$ is a truth assignment). Therefore the vertex z has percolation time at least 5, the vertex z'' has percolation time at least 6 and the vertex z''' has percolation time at least 7.

Now, suppose that $t(G) \geq 7$ and let S be any hull set of G with $t(S) \geq 7$. Note that $L \subseteq S$; also for any clause C_i, we have $U_i \cap S \neq \emptyset$ because $|N(u_{i,j}) - U_i| \leq 1$, for all i, j. This implies that $W \subseteq I^4(S)$, $U \subseteq I^6(S)$, $Y \subseteq I^5(S)$, $z \in I^6(S)$, $z'' \in I^7(S)$ and $z''' \in I^8(S)$. Furthermore, if $Y \cap I^3(S) \neq \emptyset$ then $z \in I^4(S)$, $z'' \in I^5(S)$, $z''' \in I^6(S)$ and then $t(S) \leq 6$, a contradiction. Then $Y \cap I^2(S) = \emptyset$. This implies that no pair $\{u_{i,a}, u_{i,b}\}$, where $\ell_{i,a}$ is the negation of $\ell_{j,b}$, is in S. This means that assigning true to each $\ell_{i,j}$ for which $u_{i,j} \in S$ gives us an assignment that satisfies $\mathcal{C}$.

For values $k > 7$, it suffices to subdivide the edge $z''z'''$ into a path P of length $k - 7$, appending a new leaf vertex to each vertex in P. □

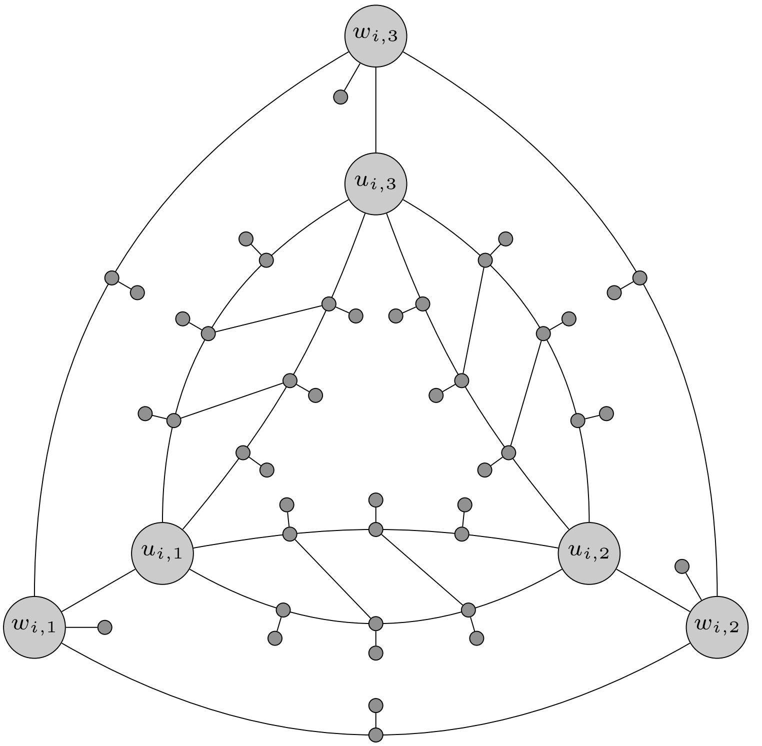

Fig. 3. Planar bipartite gadget for clause C_i.

To prove that the problem is also NP-complete for planar graphs, we make a reduction from a restricted 3-SAT, the Planar 3-SAT, which is also NP-complete [27]. The construction is similar, however, in order to maintain the planarity we need to increase the time of percolation of the so-called "*collector vertices*", which is why our proof does not work when the time is fixed.

Theorem 2.3. Percolation time *is NP-complete even when restricted to planar graphs.*

We use the following definition. If $\mathcal{C}$ is an instance of 3-SAT, then the *underlying graph of* $\mathcal{C}$ is a bipartite graph which has one vertex for each variable and one vertex for each clause and has an edge between a variable vertex and a clause vertex if and only if the corresponding clause contains a literal with the corresponding variable. If the underlying graph of $\mathcal{C}$ is planar, we say that $\mathcal{C}$ is a *planar formula*. We prove a reduction from Planar 3-SAT, which is known to be NP-complete [27], even if each variable appears in at most three clauses and every literal of the form $\bar{x}$, where x is a variable, appears in exactly one clause (see the proof of Theorem 2a in [18]).

Restricted Planar 3-SAT
Input: A 3-SAT planar formula on variables of a set X such that each variable appears in at most three clauses and every literal of the form $\bar{x}$, where $x \in X$, appears in exactly one clause.
Question: Is there a truth assignment to X that satisfies all clauses of $\mathcal{C}$?

Proof. Let $\mathcal{C}$ be an instance of Planar 3-SAT with m clauses $C_1, \ldots, C_m$ on $X = \{x_1, \ldots, x_n\}$.

We construct a graph H similar to the one obtained by Construction 1, using the gadget of Fig. 3 instead of Fig. 1. We claim that H is planar. To prove this claim, consider the underlying graph of $\mathcal{C}$ embedded into the plane. We show how to modify this embedding turning it into a graph isomorphic to H. For each clause vertex c_i, replace c_i by a copy of the gadget in Fig. 3.

Redraw each one of the three edges incident to c_i into the vertices in W_i without crossing the dashed region of the gadget, one edge for each vertex in W_i. If necessary, rename the indices in the clause gadget so that $w_{i,k}$ is adjacent to a variable vertex used in the literal $\ell_{i,k}$. Now, for each variable vertex v_j corresponding to variable x_j, contract the edge between v_j and $w_{i,k}$ such that $\ell_{i,k} = \overline{x_j}$. Finally, for each edge between vertices $w_{i,a}$ and $w_{j,b}$, subdivide this edge once and name this created vertex $y_{(i,a)(j,b)}$. The obtained graph is isomorphic to H by construction and all operations described can be made while maintaining planarity. This proves that H is planar.

From this planar drawing of H, let F be the set of faces of H that are incident to at least one vertex in Y, i.e., faces which are not bounded exclusively by clause gadget vertices. Let B be the bipartite graph with vertex set $F \cup Y$ and an edge between $f \in F$ and $y \in Y$ precisely when y is contained in the face f. Consider a spanning tree T' of B rooted at some vertex $f' \in F$. Let T be obtained from T' by deleting from T all leaves in F. For $f \in F \cap V(T)$, let $Y(f)$ be the set of children of f in T. Note that since every vertex $y \in Y$ is contained in exactly two faces, then y has at most one child in T. Recursively label the vertices in T as follows.

(1) If $y \in Y$ is a leaf in T, then let $t(y) = 3$.
(2) If $f \in F \cap V(T)$, then let $t(f) = 1 + \max\{t(y) \mid y \in Y(f)\}$.
(3) If $y \in Y$ has a child f in T, then let $t(y) = 2 + t(f)$.

We now build a planar graph G from H as follows.

(1) For each $y \in Y$ such that $t(y) > 3$, subdivide each edge incident to y to include $t(y) - 3$ new vertices each. Let $P(y)$ denote the set of subdivision vertices created this way for y.
(2) For each face $f \in F \cap V(T)$, add a vertex v_f in f. Then, for each $y \in Y(f)$, add an edge between y and v_f and then subdivide this edge to include $t(f) - t(y) - 1$ new vertices. Let $P(f)$ denote the set of subdivision vertices created this way for f.
(3) For each pair f, y such that f is a child of y in T, add two vertices adjacent to both y and v_f. Let $N(f, y)$ denote the set of these two vertices created for the pair f, y.
(4) Add one pendant vertex adjacent to each vertex created in the previous steps.

Let L be the set of all degree one vertices of G, let S be any hull set of G and note that $L \subseteq S$. For any clause C_i, we have $U_i \cap S \neq \emptyset$ as $V(G) \setminus U_i$ is convex. This implies that $W_i \subseteq I^2(S)$ and therefore $W_i \cup U_i \subseteq I^3(S)$. To analyse the remaining vertices, consider the sets defined from vertices in T recursively as follows.

(1) If $y = y_{(i,a),(j,b)} \in Y$ is a leaf in T, then let $Q(y) = \{u_{i,a}, u_{j,b}\}$ and $R(y) = \{y\}$.
(2) If $f \in F \cap V(T)$, then let $Q(f) = \cup_{y \in Y(f)} Q(y)$ and $R(f) = \{v_f\} \cup P(f) \cup_{y \in Y(f)} R(y)$.
(3) If $y = y_{(i,a),(j,b)} \in Y$ has a child f in T, then let $Q(y) = \{u_{i,a}, u_{j,b}\} \cup Q(f)$ and $R(y) = \{y\} \cup P(y) \cup N(f, y) \cup R(f)$.

We say that a vertex in $v \in V(T)$ represents the vertices in $Q(v)$ and covers the vertices in $R(v)$. If S contains at most one vertex in the set $\{u_{i,a}, u_{j,b}\}$ for $u_{i,a}, u_{j,b} \in Q(v)$ such that $\ell_{i,a}$ is the negation of $\ell_{j,b}$, then we say that v is well represented. Note that f' covers all vertices with degree at least two that are not clause vertices.

Claim 2.4. *For every $v \in V(T)$, $R(v) \subseteq I^{t(v)}(S)$. Moreover, if $v = y \in Y$ and $y \notin I^{t(y)-1}(S)$ or $v = f \in F$ and $v_f \notin I^{t(f)-1}(S)$, then v is well represented.*

Proof. We prove this by induction on the height of the subtree rooted at v in T. If $y = y_{(i,a),(j,b)} \in Y$ is a leaf in T, then $t(y) = 3$. Since $W_i \cup W_j \subseteq I^2(S)$, then $y \in I^3(S)$. If S contains both $u_{i,a}$ and $u_{j,b}$, then $y \in I^2(S)$. Therefore, if $y \notin I^2(S)$, then either $u_{i,a}$ or $u_{j,b}$ is not in S.

If $f \in F \cap V(T)$, then, by the induction hypothesis, $R(y) \subseteq I^{t(y)}(S)$ for every $y \in Y(f)$. Therefore, by following the paths with internal vertices in $P(f)$, we have $R(f) \subseteq I^{t(f)}$. If $y \in I^{t(y)-1}(S)$ for any $y \in Y(f)$, then by following the path from y to v_f with internal vertices in $P(f)$, we get $v_f \in I^{t(f)-1}(S)$. Therefore, if $v_f \notin I^{t(f)-1}(S)$, then all vertices in $Y(f)$ are well represented which implies f is well represented.

If $y = y_{(i,a),(j,b)} \in Y$ has a child f in T, then, by the induction hypothesis, $R(f) \subseteq I^{t(f)}(S)$. This fact together with the fact that $W_i \cup W_j \subseteq I^2(S)$ implies $R(y) \subseteq I^{t(y)}(S)$. If either S contains both $u_{i,a}$ and $u_{j,b}$ or $v_f \in I^{t(f)-1}(S)$, then $y \in I^{t(y)-1}(S)$. Therefore, if $y \notin I^{t(y)-1}(S)$, then either $u_{i,a}$ or $u_{j,b}$ is not in S and f is well represented which implies y is well represented. □

From Claim 2.4, we know that G has percolation time at most $t(f')$. If G has percolation time $t(f')$, then let S be a hull set that achieves this time. For any $u_{i,k} \in S \cap U$ fix a variable in X so that $\ell_{i,k}$ is true. From Claim 2.4, f' is well represented and this corresponds to a truth assignment of $\mathcal{C}$ for any value given to the unfixed variables. Now, consider $\mathcal{C}$ has a truth assignment. Let S' be obtained by choosing precisely one vertex from $u_{i,k} \in U_i$ corresponding to a true literal $\ell_{i,k}$ in C_i, for $1 \leq i \leq m$ and let $S = S' \cup L$. Since f' is well represented and by an induction similar to the one in Claim 2.4, we have that S is a hull set of G with percolation time $t(f')$. $\square$

The following questions remain open:

- Is NP-Complete the problem of deciding if $t(G) \geq 3$?
- Is NP-Complete the problem of deciding if $t(G) \geq 6$ on bipartite graphs?
- Is NP-hard the problem of determining $t(G)$ on planar bipartite graphs?

3. Polynomial time cases

In this section we present polynomial time algorithms for computing $t(G)$ for chordal graphs and trees, and deciding if $t(G) \geq t$, for $t \in \{1, 2\}$, for general graphs. It is clear that $t(G) \geq 1$ if and only if G has some vertex of degree at least 2. The next result characterizes the graphs with $t(G) \geq 2$.

Theorem 3.1. *Let G be a graph. Then $t(G) \geq 2$ if and only if there exist $u \in V(G)$ and $v \in N(u)$ such that $\overline{N}(u) \cup \{v\}$ is a hull set.*

Proof. Let $u \in V(G)$ and $v \in N(u)$ such that $\overline{N}(u) \cup \{v\}$ is a hull set. Since $u \notin I(S)$, it holds $t(G) \geq 2$. Now, let S be a hull set of G such that $t(S) \geq 2$ and let $u \in I^2(S) \setminus I^1(S)$. Hence, S contains at most one neighbour v of u. If S contains no neighbour of u, take any $v \in N(u)$ and add it to S. Then, S is a subset of $\overline{N}(u) \cup \{v\}$, which implies that $\overline{N}(u) \cup \{v\}$ is a hull set. $\square$

Now we consider chordal graphs. The following result gives a crucial tool for the determination of $t(G)$ in polynomial time for this graph class.

Theorem 3.2 ([12]). *If G is a 2-connected chordal graph, then $\{x, y\}$ is a hull set of G, for all $x, y \in V(G)$ with distance at most 2.*

Let G be a connected graph and $v \in V(G)$. Denote by $G - v$ the graph obtained by removing v from G. Recall that v is a *cut vertex of* G if $G - v$ is disconnected, and that a *block of* G is a maximal 2-connected subgraph of G. A block having at most one cut vertex is a *leaf block of* G. For each vertex u and each block B containing u, let $G_{u,B}$ be the connected component of the subgraph of G induced by $(V(G) \setminus V(B)) \cup \{u\}$ containing u, $G_{B,u}$ be the connected component of $G - u$ containing $B - u$, and $G^u_{B,u}$ the subgraph of G induced by $V(G_{B,u}) \cup \{u\}$. Denote also $S_{v,B} = S \cap V(G_{v,B})$ and $\overline{S_{v,B}} = S \setminus V(G_{v,B})$. Given sets S, T and a positive integer k, denote by $perc(S, k, T)$ the minimum i such that $|I^i(S) \cap T| \geq k$, i.e., $perc(S, k, T)$ is the minimum number of applications of the interval function on S to percolate at least k vertices of T. These definitions allow us to define some useful variations of percolation time.

- $t(u) = \max\{perc(S, 1, \{u\}) : S \text{ is hull set of } G\}$;
- $t(G) = \max\{t(u) : u \in V(G)\}$;
- $t(u, B)$ as the maximum time a hull set of $G_{u,B}$ percolates u, i.e., $t(u, B)$ is $t(u)$ on the graph $G_{u,B}$;
- $t_1(B, u)$ as the maximum time a set S of $G_{B,u}$ percolates one neighbour of u and $S \cup \{u\}$ is a hull set of $G^u_{B,u}$; and
- $t_2(B, u) = \max\{perc(S, 2, N(u)) : S \text{ is hull set of } G_{B,u}\}$.

The following equation shows how to express $t(u)$ in terms of $t_1(B, u)$ and $t_2(B, u)$. Let $B_1, \ldots, B_q$ be all blocks containing vertex u.

$$t(u) = \begin{cases} 0, & \text{if } d(u) \leq 1, \\ 1 + t_2(B_1, u), & \text{if } d(u) \geq 2 \text{ and } q = 1, \\ 1 + \min\{t_1, t_2\}, \text{ where } t_1 \text{ is the second minimum } t_1(B_j, u) & \\ \text{and } t_2 \text{ is the minimum } t_2(B_j, u), \text{ for all } j \in [q], & \text{if } q \geq 2. \end{cases} \tag{1}$$

The next lemma contains a characterization of $t_1(B, u)$.

Lemma 3.3. *Let G be a graph, u a cut vertex of G, and B a block of G containing u. Then $t_1(B, u) = \max\{s_1, s_2\}$, where $s_1 = \max\{perc(S, 1, N(u)) : S$ is a hull set of $G_{B,u}\}$ and $s_2 = \max\{t(v, B) : v \in N(u)$ for which there exists a set $S \subset V(G_{B,u})$ such that $v \in S$, S is a proper convex set of $G^u_{B,u}$, and $S \cup \{u\}$ is a hull set of $G^u_{B,u}\}$.*

Proof. Note that every set S considered in the choice of s_1 is a hull set of $G_{B,u}$. Since $V(G^u_{B,u}) = V(G_{B,u}) \cup \{u\}$, $S \cup \{u\}$ is a hull set of $G^u_{B,u}$. And that every set S considered in the choice of s_2 is a proper convex set of $G^u_{B,u}$ containing exactly one vertex of $N(u)$ such that $S \cup \{u\}$ is a hull set of $G^u_{B,u}$. In both cases, the choice is a subset $S \subseteq V(G_{B,u})$ which percolates a neighbour of u such that $S \cup \{u\}$ is a hull set of $G^u_{B,u}$, which agrees with the definition of $t_1(B, u)$.

Then, it remains to show that the set S' which matches $t_1(B, u)$ is ever recovered by the maximum between the choices of s_1 and s_2. If S' is a hull set of $G_{B,u}$, then S' is considered in the choice of s_1. Otherwise, since S' percolates at least one neighbour of u, $H(S')$ contains exactly one vertex v of $N(u)$. Define $S'' = S' \cup \{v\}$. Since $S' \cup \{u\}$ is a hull set of $G^u_{B,u}$, $S'' \cup \{u\}$ is also a hull set of $G^u_{B,u}$ containing v and, hence, S'' is considered in the choice of s_2 and in this case the set S' is equal to $S'' \setminus N(u)$. □

Denote by P_G the set of pendant vertices of the graph G and by C_B the set of cut vertices of a block B. Given two hull sets S and S', we say that $S \preceq_B S'$ if $perc(S', 1, \{v\}) \geq perc(S, 1, \{v\})$, for every $v \in V(B)$. Moreover, if $perc(S', 1, \{u\}) > perc(S, 1, \{u\})$, for at least one vertex $u \in V(B)$, then we say that $S \prec_B S'$.

We can extract almost all information about the percolation times of the vertices of $V(B)$ classifying the vertices of $(V(B)\cap S)\cup C_B$ in five classes. We will see that in some situations it is better to work with this subset of vertices than with S itself. These classes are represented by the sets $B_i(S)$, for $0 \leq i \leq 4$, defined below.

- $B_0(S) = H(P_G) \cap V(B)$;
- $B_1(S)$ is the subset of $V(B)$ contained in $I^k(S) \setminus B_0(S)$, for the minimum k, such that $B_0(S) \cup B_1(S)$ is a hull set of B.
- $B_2(S) = \{v \in C_B : S_{v,B}$ is a proper convex set of $G_{v,B}$ containing no vertex of $N[v]\}$.
- $B_3(S) = \{v \in C_B : S_{v,B}$ is a proper convex set of $G_{v,B}$ containing a neighbour of $v\}$;
- $B_4(S)$ contains the remaining vertices of C_B.

Let $v \in B_2(S) \cup B_3(S)$. Observe that $perc(S, 1, \{u\}) > perc(S, 1, \{v\})$ for every vertex $u \in V(G_{v,B}) \setminus (S \cup \{v\})$. Moreover, if $v \in B_3(S)$ and S_v is a proper convex of $G_{v,B}$ containing a neighbour of v, the set $S' = (S \setminus S_{v,B}) \cup S_v$ satisfies $perc(S, 1, \{w\}) = perc(S', 1, \{w\})$, for every $w \in V(B)$. An analogue observation can be done for $v \in B_2(S)$ and a proper convex set not containing a neighbour of v. This means that the sets $B_i(S)$, for $1 \leq i \leq 4$, can represent a great number of hull sets of G and the percolation time of the vertices of B can be determined without knowing the vertices of $S_{v,B}$, for $v \in B_2(S) \cup B_3(S)$.

Given two hull sets S and S' we say that S' *is obtained from S by moving v from B_i to B_j* if $B_i(S') = B_i(S) \setminus \{v\}$, $B_j(S') = B_j(S) \cup \{v\}$, and $B_k(S') = B_k(S)$ for $k \in [4] \setminus \{i, j\}$. In fact, $B_0(S) = B_0(S')$ for any two hull sets S and S' of G. The following lemma gives a characterization of $B_0(S)$.

Lemma 3.4. *Let G be a graph and B a block of G such that $V(B) \not\subseteq H(P_G)$. Then $H(P_G) \cap V(B) = \{v \in C_B$: there is not a proper convex set S of $G_{v,B}$ such that $S \cup \{v\}$ is a hull set of $G_{v,B}\}$.*

Proof. By Theorem 3.2, every pair of vertices of $H(P_G) \cap V(B)$ are at distance at least 3. Then $H(P_G)\cap V(B) \subseteq C_B$. Let $v \in H(P_G)\cap V(B)$. It is clear that $v \in H(V(G_{v,B})\cap P_G)$. Suppose by contradiction that $G_{v,B}$ contains a proper convex set S such that $S \cup \{v\}$ is a hull set of $G_{v,B}$. This means that $v \notin S$. Which implies that some vertex of $V(G_{v,B}) \cap P_G$ is not in S. Since every hull set of $G_{v,B}$ contains $V(G_{v,B}) \cap P_G$, we have that $S \cup \{v\}$ is not a hull set of $G_{v,B}$, a contradiction.

Conversely, let $v \in C_B$ such that $G_{v,B}$ has no proper convex set S for which $S \cup \{v\}$ is a hull set of $G_{v,B}$. Suppose by contradiction that $v \notin H(P_G \cap V(G_{v,B})) = F$. Then F is a proper convex set of $G_{v,B}$, because $v \in V(G_{v,B})$. Now, we can add to F vertices of $V(G_{v,B}) \setminus \{v\}$, one per time, until to obtain a maximal proper convex set S. This set is always found because the initial set F is a proper convex set of $G_{v,B}$ and we add a vertex $w \neq v$ to the current F only if $F \cup \{w\}$ is also a convex set. Then S is a proper subset of $V(G_{v,B})$ because it does not contain v. Then $S \cup \{v\}$ is a hull set of $G_{v,B}$, a contradiction. $\square$

The following lemma says that if we replace $S_{v,B}$ by $S' \subseteq V(G_{v,B})$ in a hull set containing $S_{v,B}$, for $v \in C_B$, obtaining a new hull set such that the percolation time of v increases, then the percolation time of the other vertices of B do not decrease under this new hull set.

Lemma 3.5. *Let G be a graph, B a block of G, and S and S' be hull sets of G such that S' is obtained by moving v from B_i to B_j, for $i \in [4]$ and $j \in [4] \setminus \{i\}$. If $perc(S', 1, \{v\}) > perc(S, 1, \{v\})$, then $S \prec_B S'$.*

Proof. Denote $t = perc(S, 1, \{v\})$. Let $u \in V(B) \setminus \{v\}$. First, consider $perc(S, 1, \{u\}) \leq t$. In this case, when v is percolated by S', u is already percolated, then $perc(S', 1, \{u\}) = perc(S, 1, \{u\})$.

Now, consider $perc(S, 1, \{u\}) > t$. We prove by induction on $k = perc(S, 1, \{u\}) - t$. If $k = 1$ and at time t of the percolation of S, the vertex u has only two neighbours percolated, being v one of them, we have that $perc(S', 1, \{u\}) > t+1$, otherwise $perc(S', 1, \{u\}) = perc(S, 1, \{u\}) = t+1$. Concluding the proof of the basis.

Next, consider $perc(S, 1, \{u\}) - t = k+1$ and suppose that for every vertex $w \in V(B)$ such that $perc(S, 1, \{w\}) - t = k'$, for $1 \leq k' \leq k$, it holds $perc(S', 1, \{w\}) \geq perc(S, 1, \{w\})$. This means that at time $k-1$ of the percolation of S, u has at most 1 percolated neighbour. By the induction hypothesis, in the percolation of S', u has at most one percolated neighbour. Then $perc(S', 1, \{u\}) \geq k+1 = perc(S, 1, \{u\})$. $\square$

The following lemma and Theorem 3.2 guarantee that, in a chordal graph, the maximum percolation times can be obtained considering only sets B_1 with at most two vertices. The following lemma also guarantees that the percolation time never decreases if a vertex is moved from $B_3 \cup B_4$ to B_2.

Lemma 3.6. *Let G be a graph, B a block of G, and S a hull set of S.*

(a) *If $perc(S, 1, \{u\}) = \max\{perc(S', 1, \{u\}) : S'$ is a hull set of $G\}$, for some $u \in V(B)$, then there is a hull set S'' satisfying $perc(S'', 1, \{u\}) = perc(S, 1, \{u\})$, such that, $B_0(S'') \cup B_1(S'') \setminus \{v\}$ is not a hull set of B, for every $v \in B_1(S'')$.*
(b) *If some vertex $v \in B_3(S) \cup B_4(S)$ can be moved to $B_2(S)$, then the set S' formed by this movement satisfies $S \preceq_B S'$.*

Proof. (a) If $B_0(S) \cup B_1(S) \setminus \{v\}$ is not a hull set of B, for every $v \in B_1(S)$, we are done. Then, suppose the contrary and let $v \in B_1(S)$ such that $B_0(S) \cup B_1(S) \setminus \{v\}$ is a hull set of B. By Lemma 3.4, there exists a proper convex set of $G_{v,B}$ containing a neighbour of v. Then we can move v to B_3 forming a hull set S'. Since $perc(S, 1, \{v\}) \geq perc(S', 1, \{v\})$, by Lemma 3.5, $S \preceq_B S'$. Next, set S equal to S' and repeat this procedure until there is no more vertices like v in $B_1(S)$.

(b) Let $v \in B_3(S) \cup B_4(S)$ for which there exists a proper convex S set $G_{v,B}$ containing no vertex of $N[v]$ such that $S \cup \{v\}$ is a hull set of $G_{v,B}$, i.e., v can be moved to $B_2(S)$. Let S' be the hull set formed by moving v to B_2. Denote $t = perc(\overline{S_{G_{v,B}}}, 2, N(v))$. It is clear that $perc(S, 1, \{v\}) \leq t+1$. However, since $perc(S', 1, \{w\}) > perc(S', 1, \{v\})$, for every $w \in V(G_{v,B}) \setminus (S \cup \{v\})$, $perc(S', 1, \{v\}) = t+1$. Then, by Lemma 3.5, $S \preceq_B S'$. $\square$

Now, we present an algorithm that, given a set B_1 such that $B_0(S') \cup B_1(S') \setminus \{v\}$ is not a hull set of B, for every $v \in B_1(S')$, finds the hull set S with $B_1(S) = B_1$ maximizing $perc(S, 1, \{u\})$, for every $u \in V(B)$. The following two conditions are useful in this algorithm. They consider a hull set S and a block B of the graph.

Condition 1. $\min\{t_1(B', v) : B' \neq B\} > perc(\overline{S_{v,B}}, 1, N(v))$.

Condition 2. $perc(\overline{S_{v,B}}, 2, N(v)) > perc(\overline{S_{v,B}}, 1, N(v))$.

Algorithm 1.

Input: a chordal graph G, a block B, and a subset $B_1 \subseteq V(B)$ such that $B_0 \cup B_1$ is a hull set of B and $B_0 \cup B_1 \setminus \{u\}$ is not a hull set of B, for every $u \in B_1$.

Define B_2 as the subset of vertices $v \in C_B \setminus (B_0 \cup B_1)$ for which there exists a proper convex set in $G_{v,B}$ containing no vertex of $N[v]$.

Initialize B_4 as an empty set and B_3 as the remaining vertices of C_B.

While there is a vertex in B_3 satisfying Conditions 1 and 2 for B and the current hull set S, **move** to B_4 the vertex v satisfying such conditions and minimizing $perc(\overline{S_{v,B}}, 1, N(v))$.

end

The correctness of Algorithm 1 is presented in the following theorem.

Theorem 3.7. *Let S be the set obtained by Algorithm 1 applied on a chordal graph G, a block B of G, and a subset $B_1 \subseteq V(B)$ such that $B_0(S) \cup B_1$ is a hull set of B and $B_0(S) \cup B_1 \setminus \{u\}$, for every $u \in B_1$, is not a hull set of B. Then there is no hull set S' of G with $B_1(S') = B_1$ such that $perc(S', 1, \{v\}) > perc(S, 1, \{v\})$, for some $v \in V(B)$.*

Proof. We prove by induction on the percolation time of the vertices of $V(B)$. It holds $perc(S, 1, \{u\}) = 0$, for every $u \in S \cap V(B)$. Let S' be any hull set of G with $B_1(S') = B_1(S)$. By the definitions of the $B_i(S')$ sets, it is easy to see that $S' \cap V(B) = B_1(S')$. Then, $S \cap V(B) = S' \cap V(B)$ and $perc(S', 1, \{u\}) = 0$, for every $u \in S \cap V(B)$. Completing the proof of the basis.

Now, let k be a nonnegative integer. Suppose that for every $v \in V(B)$ such that $perc(S, 1, \{v\}) \leq k$, there is no hull set S' with $B_1(S') = B_1(S)$ satisfying $perc(S', 1, \{v\}) > perc(S, 1, \{v\})$.

Claim 1. If $v \in B_4(S)$, then $perc(\overline{S_{v,B}}, 2, N(v)) > perc(\overline{S_{v,B}}, 1, N(v))$ and $\min\{t_1(B', v) : B' \neq B\} > perc(\overline{S_{v,B}}, 1, N(v))$. Let S^{r-1} and S^r be the hull sets of G in the beginning and in the ending of iteration r of the algorithm in which v is chosen to be moved to B_4, respectively. Denote $t = perc(\overline{S^{r-1}_{v,B}}, 1, N(v))$. Let v' be the neighbour of v in $V(B)$ such that $perc(\overline{S^{r-1}_{v,B}}, 1, \{v'\}) = t$. By Condition 1, $\min\{t_1(B', v) : B' \neq B\} > t$ and, by Condition 2, $perc(\overline{S^{r-1}_{v,B}}, 2, N(v)) > t$. Then, $perc(S^r, 1, \{v\}) > t$ and, by Lemma 3.5, $perc(\overline{S^r_{v,B}}, 1, \{v''\}) > t$, for every $v'' \in (N(v) \cap V(B)) \setminus \{v'\}$. The minimality of v at iteration r implies that the movements of the remaining iterations will interfere only in percolation times greater than t. Then $perc(\overline{S_{v,B}}, 1, \{v'\}) = t$. By Lemma 3.5 again, $perc(S, 1, \{v\}) > t$ and $perc(S, 1, \{v''\}) > t$, for every $v'' \in N(v) \cap V(B) \setminus \{v'\}$. Then the claim holds.

Let $u \in V(B) \setminus B_1(S)$ such that $perc(S, 1, \{u\}) = k + 1$ and suppose by contradiction that S' is a hull set of G with $B_1(S') = B_1(S)$ satisfying $perc(S', 1, \{u\}) > k + 1$. Recall that $B_0(S) = B_0(S')$ and $B_1(S) = B_1(S')$. Then, using Lemma 3.6(b), we can assume $B_2(S) = B_2(S')$ and $B_3(S) \cup B_4(S) = B_3(S') \cup B_4(S')$. Denote $N^k_u = N(u) \cap I^k(S)$. Since $perc(S, 1, \{u\}) = k + 1$, $|N^k_u| \geq 2$. If $|N^k_u \cap V(B)| \geq 2$, by the induction hypothesis, we would have $perc(S', 1, \{u\}) = k + 1$. Therefore, we can consider $|N^k_u \cap V(B)| \leq 1$. This implies $u \notin B_2(S)$.

If $|N^k_u \cap V(B)| = 0$, then $u \in B_0(S) \cup B_4(S)$. First, consider $u \in B_0(S)$. Then, $u \in I^{k+1}(P_G)$. Since P_G is contained in any hull set of G it holds $perc(S', 1, \{u\}) \leq k + 1$. Next, consider $u \in B_4(S)$. However, this case is also not possible, since Claim 1 implies $|N^k_u \cap V(B)| \geq 1$.

Then, we can assume from now on that $|N^k_u \cap V(B)| = 1$. In this case, N^k_u contains at least one vertex of $G_{u,B}$. First, consider $u \in B_0(S)$. Then, at least one neighbour of u contained in $G_{u,B}$ belongs to $I^k(P_G)$. Then $perc(S', 1, \{u\}) \leq k + 1$. Now, consider $u \in B_4(S)$. Since N^k_u contains only one vertex of $V(B)$, by Claim 1, $perc(S, 1, \{u\}) = \min\{t_1(B', u)\} + 1$. If $u \in B_3(S')$, we would have $perc(S', 1, \{u\}) = k + 1$, then $u \in B_4(S')$. Since $perc(S', 1, \{u\}) > perc(S, 1, \{u\})$, we have $perc(S', 1, \{u\}) = perc(\overline{S'_{u,B}}, 2, N(u)) + 1$. This implies $perc(\overline{S'_{u,B}}, 2, N(u)) > \min\{t_1(B', u)\} = r$. However, this implies that $I^r(S')$ contains two neighbours of u, which is not possible.

At last, consider $u \in B_3(S)$. Then $perc(S, 1, \{u\}) = 1 + perc(\overline{S_{u,B}}, 1, N(u))$. In this case $u \in B_4(S')$, otherwise $perc(S', 1, \{u\}) = perc(S, 1, \{u\})$. Then $perc(S', 1, \{u\}) = 1 + \min\{perc(\overline{S_{u,B}}, 2, N(u)), \min \{t_1(B', u) : B' \neq B\}\}$. Therefore, $\min\{t_1(B', u) : B' \neq B\} > perc(\overline{S_{u,B}}, 1, N(u))$. Since $|N^k_u \cap V(B)| = 1$, we have $perc(\overline{S_{v,B}}, 2, N(v)) > perc(\overline{S_{v,B}}, 1, N(v))$. Then u satisfies Conditions 1 and 2 on S, which contradicts the assumption that the algorithm terminates having no vertices of $B_3(S)$ satisfying Conditions 1 and 2 on S. □

By Lemma 3.3, Lemma 3.6(a), and Theorem 3.7, the algorithm for computing $t_1(B, u)$ and $t_2(B, u)$ for a given vertex u of a block B of a chordal graph is direct, considering that the following auxiliary data is known:

- $t(v, B)$, for every $v \in N(u) \cap C_B$,
- $t_1(B, w)$ for every $w \in C_B$, and
- what vertices of C_B belong to B_2 (Lemma 3.6(b)).

We divide the algorithm in two parts. Part 1 consists of finding $t_1(B, u)$ and $t_2(B, u)$ for every pair (u, B) where u is a cut vertex of B. Part 2 consists of finding $t_2(B, u)$ for every pair (u, B) where u is not a cut vertex of B.

Given a pair (u, B), for finding $t_1(B, u)$, for example, one can apply Algorithm 1 on graph $G^u_{B,u}$, block B, and every subset $B_1 \subseteq V(B)$, with $|B_1| \leq 2$, saving the current maximum value. The complexity of this algorithm, considering that all auxiliary data is known, is $O(|V(B)|^2|E(B)|)$, since the computation of the convex hull of a set and Algorithm 1 is $O(m)$, where $m = |E(G)|$. Now, we show how to get the auxiliary data in the moment of the consideration of the pair B, u without increasing the total time complexity of the algorithm.

First, consider the problem of identifying what vertices belong to B_2. Let B be a block of G. We say that a non cut vertex $v \in V(B)$ is a B_2-*vertex* if there is a proper convex set S in G, containing no vertex of $N[v]$, such that $S \cup \{v\}$ is a hull set of G. It is clear that $v \in C_B$ belongs to $B_2(S)$ if and only if v is a B_2-vertex of every $G^v_{B',v}$, for $B' \neq B$.

Lemma 3.8. *Let G be a graph and v a non cut vertex of G. Then v is a B_2-vertex if and only if there exists $u \in N^2(v)$ such that every $w \in N(w) \cap C_B$ belongs to $B_2(S)$, where B is the block containing v.*

Proof. The necessity is direct by the definition.

Conversely, consider $u \in N^2(v)$ and every $w \in N(w) \cap C_B$ belongs to $B_2(S)$, where B is the block containing v. Define S containing u, union the set S_w that is a certificate that $w \in N(w) \cap C_B$ belongs to $B_2(S)$, for every $w \in N(w) \cap C_B$, union a proper convex set of $G_{B',w'}$ containing a neighbour of w' for every $w' \in C_B \setminus N(w)$. □

We are ready to present the algorithm to determine what vertices of C_B belong to B_2, for every block B of a chordal graph G. Let $u \in V(G)$ be a cut vertex, and B be a block containing u. We denote by $h(u, B)$ the maximum number of blocks that one must cross to reach a leaf block, starting from u and without passing by B. Observe that initially we are able to decide only for those (u, B) with $h(u, B) = 0$ (i.e., all blocks containing u other than B are leaf blocks). After this, we can decide some other pair u, B, namely those with $h(u, B) = 1$. In general, if $h(u, B) = k$ and it is known for all pairs (u', B') with $h(u', B') = k-1$, then we can decide for (u, B). Thus, if $t = \max_{(u,B)} h(u, B)$ and $D_1, \ldots, D_t$ is a partition of the pairs (u, B) such that $(u, B) \in D_i$ if and only if $h(u, B) = i$. One can see that it is possible to decide for all (u, B) in time $O(m\alpha)$, where α it the time to check Lemma 3.8 for a vertex v, considering that it is known what vertices of C_B belong to B_2.

It remains to show how to compute all $t_1(B, u)$ and $t_2(B, u)$. We can adapt the algorithm described in the previous paragraph to incorporate Part 1 of the algorithm described above. Observe that the precedence constraints are weaker for $t_1(B, u)$ and $t_2(B, u)$ than for to decide what vertices belong to B_2, since to compute $t_1(B, u)$ and $t_2(B, u)$ we need of the auxiliary data restricted to the vertices of B, while to decide what vertices belong to B_2 we need the information of the vertices of all blocks of B', for $B' \neq B$. Completed this case, all data needed to run Part 2 is known. Then, one can see that the total time of the algorithm is $O(m(\alpha + n^2m) + n^3m)$. Resulting in a total time complexity $O(n^2m^2)$.

Theorem 3.9. *Let G be a chordal graph. If G is 2-connected, then $t(G)$ can be computed in time $O(n^2m)$; otherwise, $t(G)$ can be computed in time $O(n^2m^2)$.*

When we apply the algorithm for chordal graphs on a tree, we obtain a linear time algorithm, since every block has size exactly two, the candidate hull sets S are only 4, the ones having $B_1(S)$ equal to $\{u\}$, or $\{v\}$, or $\{u, v\}$, or the empty set.

Theorem 3.10. *If T is a tree, then $t(T)$ can be computed in linear time.*

References

[1] H. Amini, Bootstrap percolation in living neural networks, J. Stat. Phys. 141 (3) (2010) 459–475.
[2] R. Araújo, R. Sampaio, V. Santos, J.L. Szwarcfiter, The convexity of induced paths of order three and applications: complexity aspects, Discrete Appl. Math. (2015) in press.
[3] J. Balogh, B. Bollobás, Bootstrap percolation on the hypercube, Probab. Theory Related Fields 134 (4) (2006) 624–648.
[4] J. Balogh, B. Bollobás, H. Duminil-Copin, R. Morris, The sharp threshold for bootstrap percolation in all dimensions, Trans. Amer. Math. Soc. 364 (5) (2012) 2667–2701.
[5] J. Balogh, B. Bollobás, R. Morris, Bootstrap percolation in three dimensions, Ann. Probab. 37 (4) (2009) 1329–1380.
[6] J. Balogh, B. Bollobás, R. Morris, Bootstrap percolation in high dimensions, Combin. Probab. Comput. 19 (5–6) (2010) 643–692.
[7] J. Balogh, G. Pete, Random disease on the square grid, Random Struct. Algorithms 13 (1998) 409–422.
[8] F. Benevides, M. Przykucki, On slowly percolating sets of minimal size in bootstrap percolation, Electron. J. Combin. 20 (2) (2013) P46.
[9] F. Benevides, M. Przykucki, Maximum percolation time in two-dimensional bootstrap percolation, (2014) (submitted for publication). http://arxiv.org/abs/1310.4457v1.
[10] B. Bollobás, C. Holmgren, P.J. Smith, A.J. Uzzell, The time of bootstrap percolation with dense initial sets, Ann. Probab. 42 (4) (2014) 1337–1373.
[11] J. Calder, Some elementary properties of interval convexities, J. Lond. Math. Soc. 3 (1971) 422–428.
[12] C. Centeno, M.C. Dourado, L. Penso, D. Rautenbach, J.L. Szwarcfiter, Irreversible conversion of graphs, Theoret. Comput. Sci. 412 (2011) 3693–3700.
[13] G. Chae, E.M. Palmer, W. Siu, Geodetic number of random graphs of diameter 2, Australas. J. Combin. 26 (2002) 11–20.
[14] J. Chalupa, P.L. Leath, G.R. Reich, Bootstrap percolation on a Bethe lattice, J. Phys. C 12 (1) (1979) 31–35.
[15] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (3) (2009) 1400–1415.
[16] M. Chopin, Optimization problems with propagation in graphs: parameterized complexity and approximation (Ph.D. thesis), Université Paris-Dauphine, 2013.
[17] E. Coelho, M. Dourado, R. Sampaio, Inapproximability results for graph convexity parameters, in: Workshop on Approximation and Online Algorithms (WAOA-2013), in: Lecture Notes in Computer Science, vol. 8447, 2014, pp. 97–107.
[18] E. Dahlhaus, D. Johnson, C. Papadimitriou, E. Seymour, M. Yannakakis, The complexity of multiterminal cuts, SIAM J. Comput. 23 (4) (1994) 864–894.
[19] P.A. Dreyer, F.S. Roberts, Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion, Discrete Appl. Math. 157 (7) (2009) 1615–1627.
[20] P. Duchet, Convex sets in graphs. II: Minimal path convexity, J. Combin. Theory Ser. B 44 (1988) 307–316.
[21] P. Erdős, E. Fried, A. Hajnal, E.C. Milner, Some remarks on simple tournaments, Algebra Universalis 2 (1972) 238–245.
[22] M. Farber, R.E. Jamison, Convexity in graphs and hypergraphs, SIAM J. Algebr. Discrete Methods 7 (1986) 433–444.
[23] A. Fey, L. Levine, Y. Peres, Growth rates and explosions in sandpiles, J. Stat. Phys. 138 (2010) 143–159.
[24] F. Harary, J. Nieminen, Convexity in graphs, J. Diferential Geom. 16 (1981) 185–190.
[25] A.E. Holroyd, Sharp metastability threshold for two-dimensional bootstrap percolation, Probab. Theory Related Fields 125 (2) (2003) 195–224.
[26] F.W. Levi, On Helly's theorem and the axioms of convexity, J. Indian Math. Soc. 15 (1951) 65–76.
[27] D. Lichtenstein, Planar formulae and their uses, SIAM J. Comput. 11 (1982) 329–343.
[28] R. Morris, Minimal percolating sets in bootstrap percolation, Electron. J. Combin. 16 (1) (2009) R2.
[29] M. Przykucki, Maximal percolation time in hypercubes under 2-bootstrap percolation, Electron. J. Combin. 19 (2012) P41.
[30] E. Riedl, Largest minimal percolating sets in hypercubes under 2-bootstrap percolation, Electron. J. Combin. 17 (1) (2010) 13pp.
[31] M.L.J. van de Vel, Theory of Convex Structures, North-Holland, Amsterdam, 1993.