ISSN 1855-3966 (printed edn.), ISSN 1855-3974 (electronic edn.)
ARS MATHEMATICA CONTEMPORANEA 23 (2023) #P3.06
https://doi.org/10.26493/1855-3974.2340.a61
(Also available at http://amc-journal.eu)

Bootstrap percolation via automated conjecturing*

Neal Bushaw †, Blake Conka, Vinay Gupta, Aidan Kierans, Hudson Lafayette, Craig Larson, Kevin McCall, Andriy Mulyar, Christine Sullivan, Scott Taylor, Evan Wainright, Evan Wilson, Guanyu Wu

Department of Mathematics and Applied Mathematics, Virginia Commonwealth University, Richmond VA, USA

Sarah Loeb

Department of Mathematics and Computer Science, Hampden-Sydney College, Hampden-Sydney VA, USA

Received 6 June 2020, accepted 28 February 2022, published online 24 January 2023

Abstract

Bootstrap percolation is a simple monotone cellular automaton with a long history in physics, computer science, and discrete mathematics. In k-neighbor bootstrap percolation, a collection of vertices are initially infected. Vertices with at least k infected neighbors subsequently become infected; the process continues until no new vertices become infected. In this paper, we hunt for graphs which can become entirely infected from initial sets which are as small as possible. We use automated conjecture-generating software and a large group lab-based model as a fundamental part of our exploration.

Keywords: Bootstrap percolation, automated conjecturing, graph theory, percolation, cellular automata.

Math. Subj. Class. (2020): 05C35,68R05

*We thank the anonymous referees for their detailed and useful comments, suggestions, and references.
†Corresponding author.
E-mail addresses: nobushaw@vcu.edu (Neal Bushaw), conkaba@vcu.edu (Blake Conka), guptavp@vcu.edu (Vinay Gupta), kieransaf@vcu.edu (Aidan Kierans), lafayettehl@vcu.edu (Hudson Lafayette), clarson@vcu.edu (Craig Larson), mccallkj@vcu.edu (Kevin McCall), andriy.mulyar@gmail.com (Andriy Mulyar), sullivanca2@vcu.edu (Christine Sullivan), taylorsm9@vcu.edu (Scott Taylor), wainrightep@vcu.edu (Evan Wainright), wilsonea@vcu.edu (Evan Wilson), wug2@vcu.edu (Guanyu Wu), sloeb@hsc.edu (Sarah Loeb)

1 Introduction

1.1 History

Bootstrap percolation can be thought of as a graph process in which an arbitrary initial configuration of infected vertices is selected from a graph; remaining uninfected vertices with many infected neighbors are successively added to the infected set until the system stabilizes. Bootstrap percolation serves as a model of nucleation and growth [12] and has been applied in the study of crack formation [1], magnetic alloys [13], hydrogen mixtures [2], and computer storage arrays [20]. More generally, it provides an important stepping stone towards understanding other cellular automaton models with applications in physics, biology, information technology, epidemiology, and more.

The k-bootstrap model has a long and interesting history. First introduced by Chalupa, Leath, and Reich [13] in 1979 as a way to model magnetic materials, it is perhaps the simplest example of a monotone cellular automata (which were introduced by von Neumann [26], based on a suggestion of Ulam [25]). Most of the work related to bootstrap percolation has focused on finding thresholds for growing families of graphs, in which the initially infected sets are chosen at random. For the interested reader, this fascinating direction can be explored through, e.g., [3, 4, 5, 10, 11, 19]). These bootstrap models have been generalized significantly in recent years, with the advent of graph bootstrap percolation [8].

Here, however, we go in a somewhat different direction. Rather than selecting the initially infected vertices at random, we allow them to be chosen very carefully. How small could such an initial set be, given that it eventually infects the entire graph?

It is clear that if the infection can only spread to vertices with at least k infected neighbors, then such an initial infected set can contain no fewer than k vertices (otherwise, not even a single uninfected vertex can become infected). In this note, we search for those graphs which can be infected from a set *exactly* k vertices – that is, those graphs which can be infected as easily as possible.

This work is inspired by earlier results of Dairyko, Ferrara, Lidický, Martin, Pfender and Uzzell [15], and Freund, Poloczek, and Reichman [17]. These groups gave degree conditions for graphs to be infectable from a small set, in the case that $k = 2$. Similar results were later proven by Gunderson [18] and Wesolek [27] for the $k \geq 3$ cases. We stay with the $k = 2$ case, but provide non-degree based conditions for percolation from small sets.

1.2 Conjecturing

The conjectures reported below are the product of the property-relations version of the CONJECTURING program of Larson and Van Cleemput [21, 22]. While the program is described in these papers it is worth mentioning that produced conjectures are produced if they are both true for all input (graph) examples and *significant*—here this means that the produced conjecture was an improvement of either temporarily stored potential conjectures or user-supplied theoretical knowledge (theorems). The CONJECTURING program is open-source, and written to work with Sage; the code, examples, and set-up instructions are available at: `nvcleemp.github.io/conjecturing/`. A substantial effort has also been made to code graph-theoretic knowledge; this is available at: `math1um.github.io/objects-invariants-properties/`.

CONJECTURING is used a *tool* in this research; while we don't mean to add anything to the papers that describe how the program works, we will add some context for interested readers. In this paper we investigate both sufficient and necessary conditions for a graph to be 2-*bootstrap-good*. Sufficient conditions are themselves properties, often themselves boolean functions of more basic properties. The CONJECTURING program allows the user to input any list of pre-coded properties to use as "ingredients" for these sufficient conditions. These input (or basic, or pre-coded) properties have minimum complexity—or "complexity-1". A unary boolean operator, such as negation, applied to a complexity-1 property yields a complexity-2 property. The CONJECTURING will systematically build every possible property-expression from these input properties and (built-in) boolean operators. The CONJECTURING program also allows the user to input a list of graphs. A property P will be considered to be a possible sufficient condition for a graph to be 2-*bootstrap-good* if every input graph G that has property P is also 2-*bootstrap-good*. A possible sufficient condition P will only be added as a (potential) *conjecture* if it is true for some input graph G which is false for every other currently stored sufficient condition conjecture.

The user of the CONJECTURING can improve the quality of the produced conjectures by adding more pre-coded properties, and by adding as input graphs any graphs that have been found to be counterexamples to previous conjectures. The CONJECTURING program simply systematizes and automates what a human mathematician already does: a human mathematician's sufficient condition condition conjectures for a graph to be 2-*bootstrap-good* are necessarily properties "built" from graph properties she already knows and should be true at least for the specific graphs she has tested them on. In a precise sense, a human cannot make a "better" conjecture for a graph to be 2-*bootstrap-good* than the conjectures the CONJECTURING program makes (from the same inputs). Maybe the most important feature of the program is its ability to systematically consider every property up to some complexity—no human can do this.

A last feature we will mention of the CONJECTURING program is its ability to use theorems or *theoretical knowledge*. Suppose it is known that property P is a sufficient condition for a graph to be 2-*bootstrap-good*. This can be added as an input to the program: any conjectured sufficient condition property Q must be true for some input graph G which does not have property P. This feature of the program can be useful to "grow" a theory. In fact, some simple theorems may later be superseded. There is utility still in simple theorems: Dirac's Theorem, for instance, is still of interest—even though it is now implied by less-simple, more comprehensive, theorems.

1.3 Definitions

Here, we define precisely the k-bootstrap percolation model. Wherever possible, we use standard graph theoretic notation (see, e.g., [7]).

Let k be a natural number, G a graph, and let $\mathcal{I} \subseteq V(G)$ be a set of vertices which we think of as being initially *infected*. We then grow the infected set as follows: if an uninfected vertex v has at least k neighbors which are infected, then we add it to $\mathcal{I}$. That is, whenever we have a vertex $v \in V(G) \setminus \mathcal{I}$ with $|N(v) \cap \mathcal{I}| \geq k$, then we move v to $\mathcal{I}$.[1] Eventually, this process stabilizes – either every remaining uninfected vertex has fewer

[1]There is some abiguity here – we have described this process as happening a single vertex at a time. That is, each vertex of the graph in sequence checks its number of infected neighbors, and becomes infected if this is large. It is more standard to think of this infection as occurring in 'rounds', where *every* vertex with lots of infected neighbors is infected simultaneously. Because the process is monotone (infected vertices never uninfect), both versions reach the same final percolating set. We won't be concerned with things like the *time to percolate*, which

than k neighbors in $\mathcal{I}$, or every vertex has joined $\mathcal{I}$. We denote this final infected set by $\langle\mathcal{I}\rangle$. When $\langle\mathcal{I}\rangle = V(G)$, we say that G *k-percolates from* $\mathcal{I}$. When G is clear from context, we will say that $\mathcal{I}$ *k-percolates*; when k is also clear from context, we simply say that $\mathcal{I}$ *percolates*.

For a graph with more than k vertices, any set $\mathcal{I}$ which k-percolates must have $|\mathcal{I}| \geq k$; otherwise, there are not enough vertices in total for *any* uninfected vertex to join $\mathcal{I}$. With this minimum size in mind, we call a graph *k-bootstrap-good* if there is a set of size exactly k which k-percolates[2]. A graph which is not k-bootstrap-good is k-bootstrap-bad.

We define $m(G, k)$ to be the minimum size of a set $\mathcal{I}$ such that G k-percolates from $\mathcal{I}$. As such, our k-bootstrap-good graphs are those which have $m(G, k) = k$. In the rest of this paper, we focus on finding conditions related to 2-bootstrap-good graphs[3].

2 Lemmata (useful lemmas)

In this section, we note a few very simple results which we shall use frequently in the remainder of the paper. To be explicit, since we're only interested in graphs which might be 2-bootstrap-good, all theorems and conjectures following should be assumed to have the following extra conditions:

1. We focus exclusively on graphs with at least 3 vertices, since it requires two neighbors to become infected.
2. All graphs are connected. (The only disconnected graph which is 2-bootstrap-good is the graph with two isolated vertices)
3. All graphs have at most two blocks (as discussed in the following paragraph.).

Recall that a *block* in a graph G is a maximal connected subgraph with no cut vertex. We enforce the third condition above due to the following lemma.

Lemma 2.1. *If a graph is* 2*-bootstrap-good, then it has at most two blocks.*

Proof. Assume G is a connected graph with three blocks B_1, B_2, B_3. Since G is connected, the blocks are nontrivial (that is, the blocks are either K_2 or 2-connected graph). If both infected vertices are in a single block (say B_1), then at most one vertex vertex of B_2 will be infected – the cut vertex separating B_1 and B_2, if such a vertex exists. Thus B_2 will not be infected, and so the set cannot percolate.

If, instead, the infected vertices are in different blocks, then either no infection will spread, or if the two vertices are adjacent to a common cut vertex, they will infect first only that common cut vertex. This can then spread to the two blocks, but as before it will not move to the remaining block (since it cannot spread beyond the cut vertex). □

As a consequence of this, we note that in particular if G contains a cut edge between two bad subgraphs, then G is itself 2-bootstrap-bad.

We shall frequently make use the following two lemmas, which help us to decompose graphs which are k-bootstrap-good.

---

is itself a fascinating area of research, and so we shall use either the 'vertex-by-vertex' or 'rounds' perspective as we see fit.

[2]It is worth noting here that we only require the existence of a single small percolating set – not that *every* set of k vertices percolates.

[3]Thus wherever it is not stated, the reader should assume that we are discussing 2-bootstrap percolation and that a graph declared 'good' is in fact '2-bootstrap-good'.

Lemma 2.2. *If G is k-bootstrap-good and H is formed by adding a vertex v with at least k neighbors inside G, then H is also good.*

By infecting the initial percolating set $\mathcal{I}$ of size k in G, all of G will become infected, including (at least) k neighbors of v, and so v will also become infected. And so, our initial set $\mathcal{I}$ inside G actually percolates to all of H.

Lemma 2.3. *If G is an n vertex graph which is k-bootstrap-good, then it can be constructed from an $n-1$ vertex k-bootstrap-good graph G' and adding a new vertex adjacent to at least k vertices of G'.*

This is immediate – consider a minimum size infecting set, and let v be the very last vertex which becomes infected.

To be explicit, as we are only interested in graphs which might be 2-bootstrap-good, in order to avoid very long theorem statements, we really wish all of our theorems to have the following additional conditions:

1. As we stated in the introduction, we focus exclusively on graphs with at least 3 vertices.
2. All graphs are connected. (The only disconnected graph which is 2-bootstrap-good is the graph with two isolated vertices)
3. All graphs have at most two blocks (as discussed in the Lemmata).

We collect together this set of 'potentially bootstrap good' graphs in the definition below. This will allow us to simplify our theorems tremendously; rather than, e.g., "every connected chordal graph with at least three vertices and at most two blocks is 2-bootstrap-good", we can simply say "A graph in $\mathcal{G}$ which is chordal is 2-bootstrap-good".

Definition 2.4. We let $\mathcal{G}$ denote the set of all connected graphs of order at least three which have at most two blocks. We emphasize that the all large graphs which are 2-bootstrap-good are in $\mathcal{G}$.

2.1 Which graphs are bad?

In this section, we collect some easy properties of 2-bootstrap-bad graphs. While none of these results are new (and several seem to be folklore), we give their very short proofs here for completeness. The first two of these rely on the simple observation that pendant vertices must be initially infected in any percolating set.

Proposition 2.5. *A graph with at least two leaves with distinct parents is 2-bootstrap-bad.*

Again, this is straightforward – leaves can never become infected if they are not initially infected; thus any leaves must be initially infected. So, both leaves must be initially infected, and since these have distinct parents the infection does not spread.

Proposition 2.6. *Any graph with at least three leaves is 2-bootstrap-bad.*

As above, leaves must be initially infected, and there are simply too many to infect.

Proposition 2.7. *The path graph P_k of order $k \geq 4$ is 2-bootstrap-bad.*

Initially infected vertices x and y are either adjacent (and no spread happens), or not (in which case they infect exactly the vertex in between them if $d(x, y) = 2$ and no vertices otherwise).

Proposition 2.8. *The cycle C_k of order $k \geq 4$ is 2-bootstrap-bad.*

Consider a cycle $x_1x_2 \dots x_k$, with initially infected vertices x_i, x_j with $i < j$. If $j - i \in 1, 3, 4, \dots, k - 1$, then no new vertices are infected; otherwise, x_{i+1} is infected and the spread stops.

For the next results, we denote by $\overline{d}(G)$ the average degree of G, we denote the maximum average degree by $mad(G) := \max_{G' \subseteq G} \overline{d}(G')$; this is a well known graph parameter arising in chromatic theory. We will prove the following using a simple counting technique due to Riedl (who also uses *wasted* and *used* edges similar to our *usable edges* above) [23].

Theorem 2.9. *Let $\varepsilon > 0$. Then there is some $N = N(\varepsilon)$ such that every graph with $mad(G) < 4 - \varepsilon$ and $|G| > N$ is 2-bootstrap-bad.*

It is worth noting that this theorem is sharp, as is seen by the square of the cycle C_n^2 for each n; such graphs have $mad(G) = 4$ and are 2-bootstrap-good for each n. In fact, we will prove the corresponding result for the more general k-bootstrap model; this is again shown to be sharp by the k^{th} power of the cycle.

Theorem 2.10. *Let $\varepsilon > 0$. Then there is some $N = N(\varepsilon)$ such that every graph with $mad(G) < 2k - \varepsilon$ and $|G| > N$ is k-bootstrap-bad.*

Proof. Assume G is k-bootstrap-good, with vertices infected one at a time; let H_t be the graph induced by those vertices which are infected within the first t steps. Then since we initially infected k vertices, followed by one vertex at each time step, we have $|H_t| = t + k$. Further, each vertex was infected because it had at least k edges to the preceding infected vertices and so $\|H_t\| \geq kt$. Thus $\overline{d}(G) \geq \frac{kt}{t+k}$, and for t sufficiently large this is larger than $2k - \varepsilon$; this contradicts the maximum average degree condition. □

2.2 What is required to be good?

As is common in such problems, we provide only a few necessary conditions for a graph to be 2-bootstrap-good. The first of these is immediate from Lemma 2.3.

Proposition 2.11. *If G is good, then $\|G\| \geq 2(|G| - 2)$.*

The next result will be of considerable use to us later. Recall that the girth of a graph is the minimum of the cycle lengths present.

Proposition 2.12. *If G is 2-bootstrap-good and not P_3, then it has girth less than five.*

Proof. Consider two initially infected vertices u and v which percolate. Since we're assuming our graphs have at least 3 vertices, there is some vertex w which is becomes infected next – it is adjacent to both u and v. If $uv \in E(G)$, then we already have a triangle. Otherwise, if G is not $K_{1,2}$, then there is a fourth vertex which becomes infected; say x. Then x must be adjacent to two of $\{u, v, w\}$ – if it is adjacent to both v, w, we form a triangle; if it is adjacent to u, w or u, v we form a C_4. □

Note that this result shows that the Petersen graph is not 2-bootstrap-good.

2.3 What will guarantee goodness?

In this section, we provide a number of theorems giving sufficient conditions for a graph to be 2-bootstrap-good. The first of these require little to prove; however, they were the first conjectures provided by the CONJECTURING program, so we record them here for completeness.

Proposition 2.13. *Complete graphs are* 2*-bootstrap-good.*

Proposition 2.14. *Complete bipartite graphs are* 2*-bootstrap-good.*

Proof. Since $|G| > 2$, one of the bipartition classes class has at least two vertices; assume that G has bipartition (X, Y) with $|X| > 1$. Initially infect two vertices of X. Since the graph is complete bipartite, every vertex of Y is infected immediately. Then, the remaining vertices of X become infected in the next step. $\square$

Indeed, this remains true for the similar class of split graphs – those graphs whose vertex set can be partitioned into a clique and an independent set.

Theorem 2.15. *If G is a split graph with at most two blocks, then G is* 2*-bootstrap-good.*

Proof. First, notice that if the complete side has only one vertex, then the graph is a star (and thus either K_2 or $K_{1,2}$, since it has at most two blocks, and thus good.) The graph can have at most one pendant, v, which must lie in the independent set. Choosing v and any vertex of the complete graph which is not the parent of v will infect the entire graph, since the complete graph will become immediately infected and each non-pendant in the independent set must be adjacent to at least two vertices of the complete graph. If there is no pendant, then infecting any two vertices of the complete graph will suffice. $\square$

The above classes of graphs percolate very quickly (in at most 3 steps). Next, we see a class of graphs which percolates, but not necessarily in a fixed number of steps. Recall that a graph is *locally connected* if the open neighborhood of every vertex is a connected graph.

Theorem 2.16. *If a graph $G \in \mathcal{G}$ is locally connected, then it is* 2*-bootstrap-good.*

Proof. First, note that a locally connected graph has no pendants – otherwise, the neighborhood of the pendant vertex's parent contains an isolated vertex. Hence let G be a locally-connected graph, v be any vertex and w be any neighbor of v. We initially infect $\{v, w\}$. Recall that $\langle\{v, w\}\rangle$ is then the set of vertices eventually infected from $\{v, w\}$.

As the (open) neighborhood $N(v)$ is connected there is a path $w = x_1 ... x_k = u$ from w to any other vertex u in the graph $H = G[N(v)]$ induced by $N(v)$. Note that each vertex x_i in this path is necessarily a neighbor of v. Since v and w are infected and x_2 is a neighbor of both, x_2 is also infected. Similarly $x_3, ..., x_k = u$ must all be infected. So $N(v)$ is a subset of $\langle\{v, w\}\rangle$. By a symmetric argument $N(w)$ is also a subset of $\langle\{v, w\}\rangle$.

Suppose $\langle\{v, w\}\rangle$ does not equal V. Let x be any vertex in $V \setminus \langle\{v, w\}\rangle$ that is adjacent to some vertex $y \in \langle\{v, w\}\rangle$. Since $y \in \langle\{v, w\}\rangle$ and our graph is connected, there must also be a neighbor z of y in $\langle\{v, w\}\rangle$. By the reasoning above it follows that $N(y)$ must be a subset of $\langle\{v, w\}\rangle$. But then x must be in $\langle\{v, w\}\rangle$. $\square$

It is worth noting that the above proof in fact shows that if G is locally connected and pendant-free, then G 2-percolates from *any* set of two adjacent vertices.

The CONJECTURING program made several conjectures of the form that a known sufficient condition for graph Hamiltonicity is a sufficient condition for 2-bootstrap-goodness. It is a well-known result that Dirac graphs are Hamiltonian; indeed Freund, Poloczek, and Reichmann [16] proved that they are also 2-bootstrap-good. As we will use similar techniques later, we provide a short proof here.

Theorem 2.17. *If a graph in $\mathcal{G}$ is Dirac then it is* 2*-bootstrap-good.*

Proof. It is easy to check that graphs with order three with the Dirac property are 2-bootstrap-good. Assume that Dirac graphs with fewer than n vertices are 2-bootstrap-good. Let G be a Dirac graph with n vertices; so every vertex in G has degree at least $\frac{n}{2}$.

Let H be a 2-bootstrap-good subgraph of H with a maximum number of vertices. Note that no vertex in $V \setminus H$ has more than one neighbor in H, otherwise H would not be a maximum 2-bootstrap-good subgraph of G. So every vertex in $V \setminus H$ has at least $\frac{n}{2} - 1$ neighbors in $V \setminus H$. So $V \setminus H$ induces a Dirac subgraph of G. By our inductive assumption the graph $G[V \setminus H]$ is 2-bootstrap-good. Since every vertex in $G[V \setminus H]$ has degree at least $\frac{n}{2} - 1$ and the order of $G[V \setminus H]$ is no more than the order of h, it follows that both H and $V \setminus H$ have order $\frac{n}{2}$.

So G has the structure of two complete order $\frac{n}{2}$ complete sugraphs with a matching from H to $V \setminus H$. Let v be a vertex in H, v' be the vertex it is matched to in $V \setminus H$ and w be any other vertex in $V \setminus H$. It is easy to see that $\{v, w\}$ percolates G and thus G is 2-bootstrap-good. □

As a consequence, we obtain the following easy corollary.

Corollary 2.18. *If a graph in $\mathcal{G}$ is* 2*-bootstrap-good then it is either not cubic or it is Dirac.*

Proof. A graph which is both Dirac and cubic has order at most six (and no cubic graph has order seven). Hence it suffices to prove that if G is cubic with at least eight vertices, then it is not 2-bootstrap-good. We'll call an edge 'usable' at any particular step in the infection process if it has one infected endpoint and one uninfected endpoint. If an infected graph has less than two usable edges, then the infection cannot spread any further. Consider an initial set of two infected vertices in G. There are at most six usable edges leaving this set, since G is cubic. Any new infected vertex will make two edges unusable, and add at most one usable edges; thus the total number of usable edges drops by at least one with each newly infected vertex. Hence, the final number of infected vertices can be at most five, since at this point there will be at most one usable edge remaining. □

A graph with order n is *Ore* if every pair of non-adjacent vertices have degree sum at least n: being Ore is also a sufficient condition for being Hamiltonian. The CONJECTURING program also conjectured that Ore graphs are 2-bootstrap-good – this is strictly weaker than the result proven in [15], who prove that in fact degree sum at least $n - 2$ is enough.

Recall that a graph is *chordal* if it has no induced cycle of length longer than three. In order to prove that all chordal graphs are 2-bootstrap-good, we will need the following lemma.

Lemma 2.19. *Let $G \in \mathcal{G}$ be a* 2*-connected chordal graph and $S \subsetneq V(G)$ such that $|S| \geq 2$ and $G[S]$ is connected. Then there is some $x \in V(G) \setminus S$ such that x is adjacent to at least two vertices $v, w \in S$.*

Proof. Consider building an auxiliary graph G' by adding a new vertex y adjacent to everything in S. By its construction, G' is 2-connected. Pick $x \in V \setminus S$ to minimize the total length of two internally disjoint paths from x to p; call these paths $P_1 = xv_1 \dots v_n y$ and $P_2 = xw_1 \dots w_m y$. By minimality, both v_n and w_m are both in S. Since $G[S]$ is connected, there is a path from v_n to w_m in S; let $P = v_n p_1 \dots p_k w_m$ be a minimum length such path. By taking the union of $P_1 v_n$, $P_2 v_m$ and P, we find a cycle, which by minimality must be induced. But since G is chordal, this means the cycle is a triangle, and it means that x is the vertex we wanted. □

From this lemma, we easily deduce that all chordal graphs are 2-bootstrap-good.

Theorem 2.20. *If $G \in \mathcal{G}$ is chordal, then it is* 2*-bootstrap-good.*

Proof. If G contains a single block, infect any two adjacent vertices. Otherwise, infect one vertex of each block (both of which are adjacent to the cut vertex). Thus in each block we have two infected adjacent vertices, and so by repeatedly applying Lemma 2.19 we infect a new vertex as long as there is some uninfected vertex. □

In fact, something somewhat stronger is true – every chordal graph is a *strangulated graph*; this somewhat less common class of graphs consists of those graphs in which deleting the edges of any induced cycle of length greater than three would disconnect the remaining graph.

Theorem 2.21. *If $G \in \mathcal{G}$ is a strangulated graph, then it is* 2*-bootstrap-good.*

Proof. A strangulated graph can be constructed from chordal graphs and maximal planar graphs by gluing along cliques [24]. If such a graph is a block, then all the gluings occur along cliques of size at least two. We argue that any two adjacent vertices will infect the graph since both chordal graphs and maximal planar graphs can be infected from any initial adjacent pair (this is argued above for chordal graphs).

Indeed, let H be a maximal 2-bootstrap-good subgraph of a maximal planar graph G (such maximal planar graphs are well known to be triangulations). If $H \neq G$, then there must be some vertex $v \in G - H$ with a neighbor $w \in H \cap N(v)$. We orient the vertices around w clockwise as $x_1, x_2, \dots, x_k$. Note that $v = x_i$ for some $i \in [k]$, and there is at least one x_j from H (possibly with $i = j$). Hence at some point there must be a pair $x_\ell, x_{\ell+1}$ (we think cyclically, allowing x_k, x_1) with exactly of the pair in H and the other in $G - H$. But then $\{w, x_\ell, x_{\ell+1}\}$ lie on a common face, which must be a triangle. As such, any infection which percolates on H also spreads to all of $w, x_\ell, x_{\ell+1}$ contradicting maximality. Thus it must be that $H = G$, and so each triangulation can be percolated from any adjacent pair. Note that if a strangulated graph has two blocks, then there it has only one gluing that is along a single vertex; infecting a single neighbor from each adjacent block will infect the entire graph via the infection processes described above inside each block. □

A different superclass of chordal graphs is that of *dually chordal graphs* (so named because they are the clique graphs of chordal graphs, and thus dual in nature to chordal graphs). An alternate characterization is that a graph is dually chordal if and only if the hypergraph of its maximal cliques is the *dual* is a hypertree [9] (we give a more technical version of this somewhat non-standard term inside the proof). These graphs, like chordal graphs, are always 2-bootstrap-good.

Theorem 2.22. *If $G \in \mathcal{G}$ is dually chordal, then it is 2-bootstrap-good.*

Proof. We first make use of an alternate characterization. If a graph is dually chordal if the auxiliary hypergraph formed with $V(H) = V(G)$ and $E(H) = \{X : G[X]$ is a maximal clique$\}$ is a hypertree (that is, it is connected and has no cycles). If G contains a single block, then each pair of cliques intersects in at least two vertices; thus infecting any adjacent vertices will infect the entire block (as it percolates through the cliques of the instersection hypertree). For a graph with two blocks, we once again infect a single vertex of each block, with both adjacent to the cut vertex. □

Next, recall that a graph is called a *cograph* (short for complement reducible) when it contains no induced copy of the path P_4. The CONJECTURING program conjectured that such graphs are 2-bootstrap-good.

Theorem 2.23. *If $G \in \mathcal{G}$ is a cograph, then it is 2-bootstrap-good.*

Proof. Cographs can be constructed by taking disjoint unions and joins of cographs, starting from single vertices [14]. We proceed by strong induction on order of our cograph; the base cases are trivial. Consider next a cograph which is a single block. Since $G \in \mathcal{G}$, we know that G is connected and thus it arises from taking the join of two cographs G_1 and G_2. Consider infecting two vertices from G_1; this will infect all of G_2 in the next step, and these will infect the remainder of G_2 in the second step. Note that this shows something slightly stronger – we can infect any two vertices in either part of their block. Therefore, if G is constructed from two blocks G_1 and G_2 sharing a cut-vertex, and each G_i was constructed by taking the join of $H_{i,1}$ and $H_{i,2}$ with at least two vertices each, then we simply select a vertex in $H_{1,k}$ and $H_{2,j}$ which are adjacent to the cut vertex; this will infect the cut vertex, and then spread to the blocks by the preceding argument. □

3 Which Kneser graphs are good?

Finally, we make a somewhat different attack; rather than proving a general condition is sufficient, we explore a particular class of graphs and characterize those which are good. In particular, recall that the *Kneser graph* $KG_s^{(t)}$ is a graph whose vertices are the t element subsets of $[s]$, with two vertices adjacent when their corresponding subsets are disjoint. Trivially, this graph is an independent set whenever $s < 2t$. But when is it 2-bootstrap-good?

$KG_1^{(1)}$ and $KG_2^{(1)}$ are both trivially 2-bootstrap-good, and these are the only interesting Kneser graphs with $s \leq 2$. Further, we note that for $k \geq 2$, the graph $KG_{2t}^{(t)}$ is a collection of disjoint edges; this is clearly not 2-bootstrap-good, so we may assume that $s \geq 2t + 1$. All remaining possibilities are covered by the following theorem.

Theorem 3.1. *Assume $s \geq 3$. A Kneser graph $KG_s^{(t)} \in \mathcal{G}$ is 2-bootstrap-good if and only if $s \geq \min\{3t, 2t + 3\}$.*

Proof of Theorem 3.1. *Necessity*: Assume that $s < 3t$, that $s \leq 2t + 2$, and let v, w be vertices of $KG_s^{(t)}$ (that is, v and w are size t subsets of $[s]$) with which our infection begins. Note that since v and w are t element sets, we have $|v \cap w| \in [0, t-1]$. Let $A := v \cup w \subseteq [s]$, and let $B := [s] \setminus A$. Note that if a vertex x is adjacent to both v and w (that is, x can be infected by $\{v, w\}$), then x must be disjoint from A – and thus $x \subseteq B$.

Since $s \geq 3$, we have at least three vertices in $KG_s^{(t)}$, and so if $|B| < t$, then there are no vertices disjoint from $|A|$ and so v and w cannot infect any vertices. So, if our infection is to percolate we must choose v and w in order to guarantee $|B| \geq t$, and so $|A| = s - |B| \leq t + 2$. Now we need only consider two cases – either $|A| = t + 1$ (and so v and w share $t - 1$ common elements) or $|A| = t + 2$ (and so v, w share $t - 2$ common elements). Most of our work will lie in proving the first case; the second will fall shortly after.

Assume $|A| = t + 1$; then $|B| = t$ or $|B| = t + 1$. If $|B| = t$, then there is only a single vertex x, which will become infected by v and w. Since $s < 3t$, there are no vertices adjacent to both x and v or to both x and w. Thus the infection stops at precisely three vertices, and since $\binom{s}{t} \geq 4$ for all s, t satisfying our conditions, this is not the entire graph.

If $|B| = t + 1$, then similarly v and w can infect $t + 1$ new vertices. At the next stage, any two of these vertices will infect all vertices disjoint from B – these are precisely $X = \{y : |y| = t \text{ and } y \subseteq A\}$. In these last two steps, we've built a complete bipartite infected $K_{t+1,t+1}$. We will show that the infection can spread no further.

Let X be as above, and let Y be the corresponding vertices from B. Since $s < 3t$, there is again no vertex adjacent to both a vertex from X and a vertex from Y. Further, any two vertices $a, b \in X$ (or both in Y) contain $t - 1$ common elements, so between them they both contain all $t + 1$ elements of A (or of B). Then, the only vertices adjacent to both a, b are those vertices in the other half of our bipartite graph – which are already infected. Again, the infection process must stop. Since vertices with some elements from A and some elements from B are not yet infected, the initial set has not percolated.

Finally, assume $|A| = t + 2$. Then, $|B| = t$ and there is only one vertex infected by v and w. As above, the infection cannot spread any further, and since there are more than three vertices in $KG_s^{(t)}$ we cannot infect the whole graph.

Sufficiency: Suppose we have $s \geq 3$, along with $s \geq 3t$ or $s \geq 2t + 3$. We note that there are only two Kneser graphs for which $s \geq 3t$ but $s < 2t + 3$ – these are $KG_3^{(1)}$ and $KG_6^{(2)}$. Since $KG_3^{(1)} \cong K_3$, this is trivially 2-bootstrap-good. Further, one can easily check that $\{\{1, 2\}, \{2, 3\}\}$ percolates in $KG_6^{(2)}$ (as does any other pair of vertices sharing a common element). For all other values of our parameters, we may assume $s \geq 2t + 3$ (since $s \geq 3t$ will guarantee this).

As in $KG_6^{(2)}$ we choose two vertices v and w with $|v \cap w| = t - 1$. Then, letting $A := v \cup w \subseteq [s]$ and $B := [s] \setminus A$ as before, we have $|A| = t + 1$ and $|B| \geq t + 2$. Now, we partition the vertices x of $KG_s^{(t)}$ according to the size of $|A \cap x|$, noting that $|B \cap x| = t - |A \cap x|$. We denote these sets $A_0, A_1, \ldots, A_t$, where $|A_i| = i$.

Initially, we infect vertices v and w. In the second round, v and w infect all vertices disjoint from A; that is, all those vertices in A_0. These vertices in A_0 then infect all those vertices in A_t (which are disjoint from B). Since $|B| \geq t + 2$, we can choose two vertices b_1 and b_2 in A_0 which overlap which share $t - 1$ elements, so that $|b_1 \cup b_2| = t + 1$. Then, there will be at least one element of $y \in B \setminus (b_1 \cup b_2)$, so b_1 and b_2 can infect the vertices in A_{t-1}. Finally, we show that our infection percolates from this point.

Claim 3.2. *If all vertices in A_t, A_{t-1}, and A_0 are infected, the entire graph will become infected.*

Proof of Claim 3.2. For any choice of $k \in [1, t]$ we can choose two vertices of A_k such that we can from them infect a vertex in A_{t-k} and in A_{t-k+1}. Choose two vertices v and

w from A_k for which $|v \cap A| = |w \cap A|$ and $|(v \cap B) \cap (w \cap B)| = t - k - 1$. Now, since $|A| = t + 1$, we have $t + 1 - k$ elements of A at our disposal, and since $|B| \geq t + 2$ we have at least $t + 2 - (t - k + 1) = k + 1$ elements available from B. Since a vertex of A_{t-k+1} requires $t - k + 1$ elements of A and $t - (t - k + 1) = k - 1$ elements of B, and so such a vertex exists. Further, since we have every element of A_k already infected, we can choose $v \cap A$ and $w \cap A$ such that any $t + 1 - k$ elements are available, and since our choice of $v \cap B$ and $w \cap B$ is independent of our choice these, we can choose $v \cap B$ and $w \cap B$ so that any $k + 1$ entries are available from B; this allows us to infect our desired vertex in A_{t-k+1}.

To infect a vertex in A_{t-k}, choose v and w such that $|(v \cap A) \cap (w \cap A)| = k - 1$ and such that $v \cap B = w \cap B$. Then we have again $t + 1 - (k + 1) = t - k$ elements available from A, and at least $t + 2 - (t - k) = k + 2$ elements of B. Thus we can find a vertex in A_{t-k} adjacent to both v and w, and which will thus become infected. As before, we can infect any vertex of A_{t-k} this way. □

End of Proof of Theorem 3.1. □

4 Conclusion and further work

This is an introductory exploration to the area of very small percolating sets. Building on the work of Dairkyo et al. [15], and others, we used the automated conjecturing framework to begin a systematic search for classes of graphs which are 2-bootstrap-good (or 2-bootstrap-bad). From this starting point, we've given a number of not-so-hard-to-prove but quite-hard-to-discover conditions (both necessary and sufficient) for a graph to be 2-bootstrap-good. It remains an intriguing open question to find a full characterization of such graphs (however, at this early stage we do not even have a conjecture of what such a characterization might look like).

Further, there are many natural generalizations of these results to explore. In particular, what properties will guarantee that a graph has a k-element percolating set in k-bootstrap percolation? This paper explores the $k = 2$ case, but $k = 3$ and higher are as interesting. In addition, bootstrap percolation is just one of many monotone cellular automata which one can define on a graph (as a group, these are all examples of *graph bootstrap percolation* defined in the 1960s by Bollobás under the name *weak saturation* [6]). What graphs have the smallest possible percolating sets in these more general models?

Finally we report a conjecture that attracted our interest but which we did not resolve. The *diameter* of a (connected) graph is the maximum distance between any pair of its vertices. Notice that a graph with diameter no more than two has at most two blocks. A graph is *perfect* if the chromatic number and clique number of every subgraph is equal. This class of graphs includes, for instance, bipartite graphs and chordal graphs. As such, there is relation between this conjecture and Theorem 2.20.

Conjecture 4.1. *If a graph in $\mathcal{G}$ is perfect and its diameter is no more than two then the graph is* 2*-bootstrap-good.*

This conjecture was produced by the CONJECTURING program and, like the proved conjectures reported above, only guaranteed to be true for the input graphs used when the program was run. It is a surprising fact that many conjectures of the program are in fact true.

References

[1] J. Adler and A. Aharony, Diffusion percolation. I. Infinite time limit and bootstrap percolation, *J. Phys. A* **21** (1988), 1387–1404, doi:10.1088/0305-4470/21/6/015, https://doi.org/10.1088/0305-4470/21/6/015.

[2] J. Adler and U. Lev, Bootstrap Percolation: visualizations and applications, *Brazilian J. Phys.* **33** (2003), 641–644, doi:10.1590/S0103-97332003000300031, https://doi.org/10.1590/S0103-97332003000300031.

[3] M. Aizenman and J. L. Lebowitz, Metastability effects in bootstrap percolation, *J. Phys. A* **21** (1988), 3801–3813, doi:10.1088/0305-4470/21/19/017, https://doi.org/10.1088/0305-4470/21/19/017.

[4] J. Balogh, B. Bollobás, H. Duminil-Copin and R. Morris, The sharp threshold for bootstrap percolation in all dimensions, *Trans. Am. Math. Soc.* **364** (2012), 2667–2701, doi:10.1090/s0002-9947-2011-05552-2, https://doi.org/10.1090/s0002-9947-2011-05552-2.

[5] J. Balogh, B. Bollobás and R. Morris, Bootstrap percolation in three dimensions, *Ann. Probab.* **37** (2009), 1329–1380, doi:10.1214/08-aop433, https://doi.org/10.1214/08-aop433.

[6] B. Bollobás, Weakly k-saturated graphs, in: *Beiträge zur Graphentheorie (Kolloquium, Manebach, 1967)*, Teubner, Leipzig, pp. 25–31, 1968.

[7] B. Bollobás, *Modern Graph Theory*, volume 184 of *Graduate Texts in Mathematics*, Springer-Verlag, New York, 1998, doi:10.1007/978-1-4612-0619-4, https://doi.org/10.1007/978-1-4612-0619-4.

[8] B. Bollobás, P. Smith and A. Uzzell, Monotone cellular automata in a random environment, *Comb. Probab. Comput.* **24** (2015), 687–722, doi:10.1017/s0963548315000012, https://doi.org/10.1017/s0963548315000012.

[9] A. Brandstädt, F. Dragan, V. Chepoi and V. Voloshin, Dually chordal graphs, *SIAM J. Discrete Math.* **11** (1998), 437–455, doi:10.1137/s0895480193253415, https://doi.org/10.1137/s0895480193253415.

[10] R. Cerf and E. Cirillo, Finite size scaling in three-dimensional bootstrap percolation, *Ann. Probab.* **27** (1999), 1837–1850, doi:10.1214/aop/1022677550, https://doi.org/10.1214/aop/1022677550.

[11] R. Cerf and F. Manzo, The threshold regime of finite volume bootstrap percolation, *Stochastic Process. Appl.* **101** (2002), 69–82, doi:10.1016/s0304-4149(02)00124-2, https://doi.org/10.1016/s0304-4149(02)00124-2.

[12] R. Cerf and F. Manzo, Nucleation and growth for the Ising model in d dimensions at very low temperatures, *Ann. Probab.* **41** (2013), 3697–3785, doi:10.1214/12-aop801, https://doi.org/10.1214/12-aop801.

[13] J. Chalupa, G. R. Reich and P. L. Leath, Inverse high-density percolation on a Bethe lattice, *J. Statist. Phys.* **29** (1982), 463–473.

[14] D. G. Corneil, H. Lerchs and L. S. Burlingham, Complement reducible graphs, *Discrete Appl. Math.* **3** (1981), 163–174, doi:10.1016/0166-218x(81)90013-5, https://doi.org/10.1016/0166-218x(81)90013-5.

[15] M. Dairyko, M. Ferrara, B. Lidický, R. Martin, F. Pfender and A. J. Uzzell, Ore and chvátal-type degree conditions for bootstrap percolation from small sets, 2016, arXiv:1610.04499 [math.CO].

[16] D. Freund, M. Poloczek and D. Reichman, Contagious sets in dense graphs, 2015, arXiv:1503.00158 [math.CO].

[17] D. Freund, M. Poloczek and D. Reichman, Contagious sets in dense graphs, *Eur. J. Comb.* **68** (2018), 66–78, doi:10.1016/j.ejc.2017.07.011, https://doi.org/10.1016/j.ejc.2017.07.011.

[18] K. Gunderson, Minimum degree conditions for small percolating sets in bootstrap percolation, 2017, arXiv:1703.10741 [math.CO].

[19] A. E. Holroyd, Sharp metastability threshold for two-dimensional bootstrap percolation, *Probab. Theory Relat. Fields* **125** (2003), 195–224, doi:10.1007/s00440-002-0239-x, https://doi.org/10.1007/s00440-002-0239-x.

[20] S. Kirkpatrick, W. W. Wilcke, R. B. Garner and H. Huels, Percolation in dense storage arrays, *Phys. A* **314** (2002), 220–229, doi:10.1016/s0378-4371(02)01153-6, https://doi.org/10.1016/s0378-4371(02)01153-6.

[21] C. E. Larson and N. Van Cleemput, Automated conjecturing I: Fajtlowicz's Dalmatian heuristic revisited, *Artif. Intell.* **231** (2016), 17–38, doi:10.1016/j.artint.2015.10.002, https://doi.org/10.1016/j.artint.2015.10.002.

[22] C. E. Larson and N. Van Cleemput, Automated conjecturing III: Property-relations conjectures, *Ann. Math. Artif. Intell.* **81** (2017), 315–327, doi:10.1007/s10472-017-9559-5, https://doi.org/10.1007/s10472-017-9559-5.

[23] E. Riedl, Largest and smallest minimal percolating sets in trees, *Electron. J. Comb.* **19** (2012), research paper p64, 18, doi:10.37236/2152, https://doi.org/10.37236/2152.

[24] P. D. Seymour and R. W. Weaver, A generalization of chordal graphs, *J. Graph Theory* **8** (1984), 241–251, doi:10.1002/jgt.3190080206, https://doi.org/10.1002/jgt.3190080206.

[25] S. Ulam, Random processes and transformations, in: *Proceedings of the International Congress of Mathematicians, Vol. 2, Cambridge, Mass., 1950*, Amer. Math. Soc., Providence, R. I., 1952 pp. 264–275.

[26] J. von Neumann, *Theory of Self-Reproducing Automata*, University of Illinois Press, Champaign, 1966.

[27] A. Wesolek, Bootstrap percolation in Ore-type graphs, 2019, arXiv:1909.04649 [math.CO].