Bond percolation does not simulate site percolation

Nikita Gladkov*[1] and Aleksandr Zimin†[2]

[1]Department of Mathematics, UCLA, Rényi institute
[2]Department of Mathematics, MIT

March 14, 2024

Abstract

We show that a site percolation is a stronger model than a bond percolation. For this end we use the van den Berg – Kesten (vdBK) inequality to prove it for a neighborhood of a vertex with degree 3 and develop a decision tree technique to prove it for a neighborhood of a vertex with degree 2.

1 Introduction

Assume we have a graph $G = (V, E)$ and run a Bernoulli bond percolation on it with every *edge* $e \in E$ having its own probability p_e of being open independently on other edges. A Bernoulli site percolation, in contrast, is a process where every *vertex* $v \in V$ has a probability of being open. One can ask many questions about probabilistic properties of clusters connected via open vertices and edges.

To motivate our problem, recall Exercise 3.4 in [6]: "Show that bond percolation on a graph G may be reformulated in terms of site percolation on a graph derived suitably from G." Here is a formal definition.

Definition 1.1. One says that vertices v and u are connected (belong to the same cluster) in a site/bond percolation if there is a path between them passing only through open sites/bonds. In case of site percolation we also require that v and u are open themselves.

We say that a site/bond percolation μ_1 on graph $G' = (V', E')$ *simulates* a bond/site percolation μ_2 on graph $G = (V, E)$ if there is a function $f : V \to V'$ such that for all events "v is connected to u" and their boolean combinations, their probability in μ_1 is equal to the probabilities of the events $f(v)$ is connected to $f(u)$ in μ_2 and their corresponding boolean combinations.

Remark 1.2. By this definition, the simulation preserves the events like "At least n out of m vertices $v_1, \ldots, v_m$ are in the same cluster", but is not guaranteed to preserve the probability of "There is a path from a to b avoiding vertex c".

Then the solution to the exercise is given by the following theorem [3, 4].

Theorem 1.3. *For every graph G and a bond percolation μ_b on it there exists a graph G' with a site percolation μ_s which simulates μ_b.*

Proof. Let G' be a copy of G with an additional auxiliary vertex in the middle of each edge. Make all the original vertices open with probability 1 and every auxiliary vertex in the middle of each edge e open with probability p_e. This site percolation on G' is precisely isomorphic to the bond percolation on G. □

*gladkovna@ucla.edu
†azimin@mit.edu

Similarly, it is natural to ask whether site percolation can be simulated in bond percolation. Fisher [3] noted that the other direction can not be true since the argument proving Theorem 1.3 is only invertable for line graphs. We make his argument precise in Theorem 2.4 proved in Section 2. But the question becomes more interesting if we consider approximate simulations.

Definition 1.4. We say that a sequence of site (bond) percolations $\{\mu_i\}$ on graphs $G_i = (V_i, E_i)$ *approximately simulates* a bond (site) percolation ν on graph $G = (V, E)$ if there are functions $f_i : V \to V_i$ such that for all events "v is connected to u" and their boolean combinations, their probabilities in μ_i tend to the probabilities of the events "$f(v)$ is connected to $f(u)$" in ν and their corresponding Boolean combinations.

The main results of the paper are the following theorems:

Theorem 1.5. *One cannot approximately simulate site percolation on the complete bipartite graph $K_{1,3}$ (claw graph) using bond precolation.*

This is the corollary of Theorem 3.2.

Theorem 1.6. *One cannot approximately simulate site percolation on the path of length 2 using bond precolation.*

Similarly, it is the corollary of Theorem 4.6. This proof requires new inequalities concerning connectedness events in percolation. We use computer search techniques to discover new inequalities, including inequalities (5) and (6) in Section 5.

2 Preliminary remarks

Definition 2.1. A full hyperedge Bernoulli percolation is a random model on a hypergraph $H = (V, E)$. Every hyperedge e has a probability p_e of being open. One says vertices v and u are connected if there is a path between them such that each edge in the path is a subset of some open hyperedge.

Let us show that full hyperedge percolation is equal in power to the site percolation.

Theorem 2.2. *Full hyperedge percolation simulates site percolation and vice versa.*

Proof. To simulate full hyperedge percolation in site percolation, for every hyperedge with probability p_e we add one additional vertex with probability p_e and connect it to all elements of the hyperedge. All the original vertices stay open with probability 1.

Conversely, to simulate the site percolation in full hyperedge percolation, we first consider a graph G' from the proof of Theorem 1.3. By the construction, vertices of G' can be original or auxiliary. For each original vertex v we add a hyperedge e_v with probability p_v, connecting v and all adjacent auxiliary vertices. It is easy to see that all connectivity events are preserved by these simulations. □

Note that the hypergraph percolation in the sense of [9] is more general than our full hypergraph percolation and is capable of modeling more phenomena.

So, simulating site percolation is equivalent to simulating full hyperedge percolation. It is easy to see that bond percolation cannot simulate *exactly* even a hyperedge of size 3 with probability $0 < p < 1$, thus proving Fisher's remark. Indeed, a model of such a hyperedge would be some graph G with bond percolation on it.

Definition 2.3. We denote by $v_{11}v_{12}\dots v_{1i_1}|v_{21}\dots v_2 i_2|v_{n1}\dots v_{ni_n}$ the event that vertices $v_{11}, \dots, v_{1i_1}$ belong to the same cluster, vertices $v_{21}, \dots, v_{2i_2}$ belong to the same cluster, ..., vertices $v_{n1}, \dots, v_{ni_n}$ belong to the same cluster, and, moreover, these clusters are different. By $\mathbf{P}(v_{11}v_{12}\dots v_{1i_1}|v_{21}\dots v_2 i_2|v_{n1}\dots v_{ni_n})$ we denote the probability of this event in the underlying bond percolation. In particular, $\mathbf{P}(abc)$ denotes the probability of a, b and c being in the same component and $\mathbf{P}(a|b|c)$ – the probability of a, b and c being in 3 different components.

Theorem 2.4. *For all $0 < p < 1$, every simple graph $G = (V, E)$ and vertices $a, b, c \in V$ one has either $\mathbf{P}(abc) < p$ or $\mathbf{P}(a|b|c) < 1 - p$, where $\mathbf{P}$ is taken over random subgraphs given by a bond percolation on G.*

Proof. Remove all edges in G that have a probability of 0 and contract all edges with a probability of 1. Now, none of the edges in G are certain.

First, assume that there is a path P from a to b not passing through c. Then there is a nonzero probability that all edges of P will be open and the remaining edges will be closed, so $\mathbf{P}(ab|c) > 0$, where $\mathbf{P}(ab|c)$ is the probability of a and b being in the same component and c in the other. This contradicts the equation $\mathbf{P}(abc) + \mathbf{P}(a|b|c) = 1$, so every path from a to b should pass through c.

Similarly, every path from a to c should pass through b, but that means that there are no paths from a to b or c, since any such path will first go to b or c. Thus $p = 0$, which contradicts the assumptions. □

By this theorem, it is impossible to simulate the hyperedge percolation in bond percolation, we consider a question if it is possible to have an arbitrary good approximation.

Question 2.5. *For given k, $p \notin \{0, 1\}$ and $\varepsilon > 0$, does there exist a graph $G = (V, E)$, containing vertices x_1, …, x_k and a bond percolation on it with $\mathbf{P}(x_1 x_2 \dots x_k) > p - \varepsilon$ and $\mathbf{P}(x_1|x_2|\dots|x_k) > 1 - p - \varepsilon$?*

In Section 3 we note that approximate simulation is impossible for $k \geq 4$ using a lemma due to Hutchcroft [7], thus proving Theorem 1.5. Finally, we develop a new technique using decision trees to resolve the question even for 3-hyperedge (thus proving Theorem 1.6) in Section 4.

3 Simulating k-hyperedge for $k \geq 4$

In [7], essentially, the following theorem is proved using the vdBK inequality, where K_u is the cluster containing vertex u, and for each finite subset $\Lambda \subseteq V$

$$|K_{max}(\Lambda)| = \max\{|K_v \cap \Lambda| : v \in V\} = \max\{|K_v \cap \Lambda| : v \in \Lambda\}$$

is the maximal number of vertices from Λ belonging to the same cluster.

Theorem 3.1 ([7], Theorem 2.3)**.** *Let $G = (V, E)$ be a countable graph and let $\Lambda \subseteq V$ be finite and non-empty. Then for Bernoulli bond percolation one has*

$$\mathbf{P}(|K_{max}(\Lambda)| \geq 3^k \lambda) \leq \mathbf{P}(|K_{max}(\Lambda)| \geq \lambda)^{3^{k-1}+1} \tag{1}$$

and

$$\mathbf{P}(|K_u \cap \Lambda| \geq 3^k \lambda) \leq \mathbf{P}(|K_{max}(\Lambda)| \geq \lambda)^{3^{k-1}} \mathbf{P}(|K_u \cap \Lambda| \geq \lambda) \tag{2}$$

for every $\lambda \geq 1$ (not necessarily integer), integer $k \geq 0$ and $u \in V$.

This allows to prove that one can't even approximately simulate the 4-hyperedge.

Theorem 3.2. *For all $0 < p < 1$ there exists an $\varepsilon > 0$ such that for any graph $G = (V, E)$ and vertices $a, b, c, d \in V$ one has either $\mathbf{P}(abcd) < p - \varepsilon$ or $\mathbf{P}(a|b|c|d) < 1 - p - \varepsilon$.*

Proof. Assume that such a graph G exists. Let Λ be $\{a, b, c, d\}$. Then from (1) taking $\lambda = \frac{4}{3}$ one has

$$\mathbf{P}(abcd) \leq \mathbf{P}(ab \cup ac \cup ad \cup bc \cup bd \cup cd)^2.$$

If the statement of the theorem were false, this implies

$$p - \varepsilon \leq (p + \varepsilon)^2,$$

which is false for small ε. □

Note that this proves Theorem 1.5 on the spot.

4 Simulating 3-hyperedge: human proof

Now we see that it is impossible to even approximately simulate site percolation with bond percolation for the claw graph, as promised in Theorem 1.5. To prove Theorem 1.6, we need the following lemma.

Definition 4.1. For two configurations $C_1, C_2 \in \Omega = 2^{[E]}$ and a set $S \subseteq E$ we denote by $C_1 \to_S C_2$the configuration which coincides with C_1 on S and C_2 on its complement $\overline{S}$.

Lemma 4.2. *Consider two independent bond Bernoulli percolations C_1 and C_2 having the same distribution μ on the same graph G. Let a decision tree T select each edge and reveal it in both C_1 and C_2. Furthermore, allow on each step, before revealing, decide if this edge will go to the set S (thus dependent on C_1 and C_2) or to its complement $\overline{S}$. Then $C_1 \to_S C_2$ is independent of $C_2 \to_S C_1 = C_1 \to_{\bar{S}} C_2$ and both of them are distributed as μ.*

Example 4.3. If the graph is a path of length 2 from a to b, then the tree T on Figure 1 builds a set S of all edges with one end in the component of a in C_1.

Proof of Lemma 4.2. For finite graphs with $|E| = n$ and for every pair of configurations C_3, C_4 there is only one path in any decision tree leading to $C_1 \to_{S(C_1,C_2)} C_2 = S_3$ and $C_1 \to_{\bar{S}(C_1,C_2)} C_2 = S_4$ and the probability of this path is equal to $\mathbf{P}(C_1)\mathbf{P}(C_2)$, which is equal to $\mathbf{P}(C_3)\mathbf{P}(C_4)$ since the probability in Bernoulli percolation is a product of probabilities for individual edges. □

Example 4.4. For example, one can take T querying all the edges from the vertices already known to connect to the vertex a in C_1. It will assign all these edges to S and then discover the remaining edges assigning them to $\bar{S}$. Then S will be the set of all open and closed edges with at least one edge in the component of a. Lemma 4.2 claims that rerunning the choice for these edges will result in measure μ and rerunning the choice for the remaining edges will also result in measure μ.

Remark 4.5. Notice that Markov Chains method from [2] is based on the fact that rerunning the choice for edges in $\overline{S}$ preserves the measure restriction $\mu|_{a|b}$.

In our notation, it means that for $A = a|b$ and any B one has

$$\mathbf{P}(C_1 \in A \text{ and } C_1 \to_{S(C_1)} C_2 \in B) = \mathbf{P}(A \cap B)) \tag{3}$$

Theorem 4.6. *For any $p \notin \{0, 1\}$ there is some $\varepsilon > 0$ such that there is no graph $G = (V, E)$ and vertices $a, b, c \in V$ such that $\mathbf{P}(abc) > p - \varepsilon$ and $\mathbf{P}(a|b|c) > 1 - p - \varepsilon$.*

Proof. We will need multiple sets S_i for our purpose. So, we define sets S_1, S_2 and S_3, which are somewhat complex.

To build S_1, we query all edges connected to b and put them in S. Then we query all not queried edges connected to a (this is vacuous if a was connected to b) and put them in $\bar{S}$. Then we query all not queried edges connected to c and put them in S. Finally, we put the rest of the edges in $\bar{S}$. If we denote by Com_x the set of vertices, connected to x via edges open in C_1, we get

$$S_1 = \begin{cases} E \cap \left(Com_b \times \overline{Com_a} \cup Com_c \times V\right) \text{ if } C_1 \in a|b|c; \\ E \cap \left(Com_b \times V\right) \text{ if } C_1 \in abc; \\ E \cap \left(Com_b \times \overline{Com_a}\right) \text{ if } C_1 \in a|bc; \\ E \cap \left(Com_b \times \overline{Com_a}\right) \text{ if } C_1 \in ac|b; \\ E \cap \left((Com_b \cup Com_c) \times V\right) \text{ if } C_1 \in ab|c. \end{cases}$$

The only case we will actually use is $a|b|c$. S_2 is defined analogously with b and c interchanged.

$$S_2 = \begin{cases} E \cap \left(Com_c \times \overline{Com_a} \cup Com_b \times V\right) \text{ if } C_1 \in a|b|c; \\ \text{Something else otherwise.} \end{cases}$$

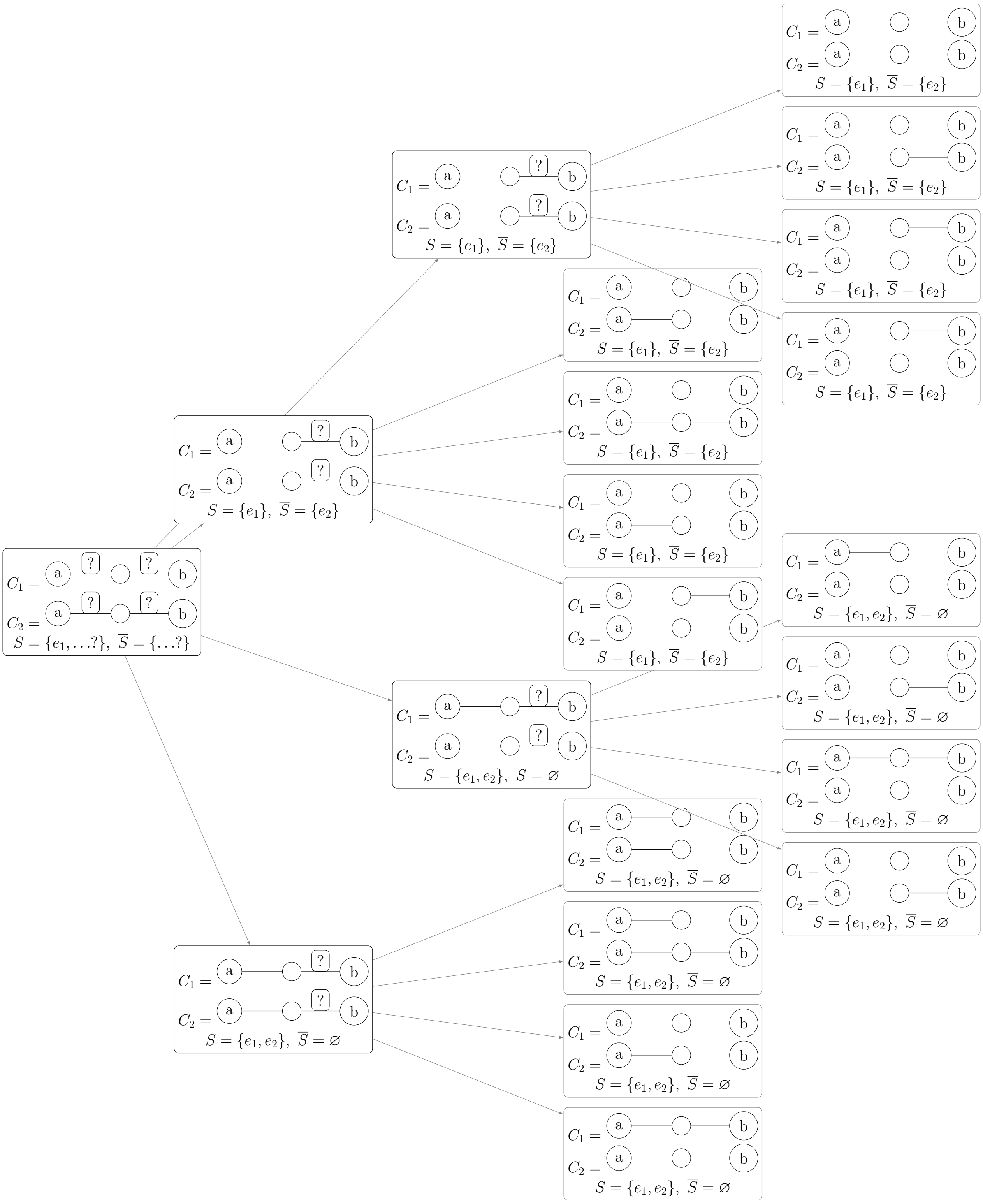

Figure 1: T corresponding to the Example 4.3

Finally, for S_3 we put all edges connected to a in $\bar{S}$, all not queried edges connected to b or c to S and the rest of the edges to $\bar{S}$.

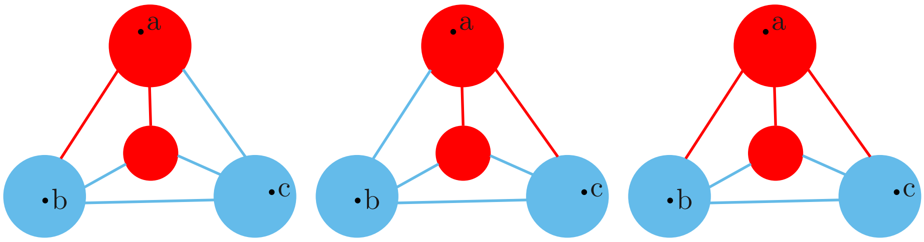

Figure 2: S_1, S_2 and S_3 for the case $C_1 \in a|b|c$. Regions surrounding a, b, c depict Com_a, Com_b and Com_c. Respective sets are in blue and their complements are in red.

$$S_3 = \begin{cases} E \cap \big((Com_b \cup Com_c) \times \overline{Com_a}\big) \text{ if } C_1 \in a|b|c; \\ \varnothing \text{ if } C_1 \in abc; \\ \text{Something else otherwise.} \end{cases}$$

The key observation will be that when $C_1 \in a|b|c$ and $C_1 \to_{S_3} C_2 \in ab \cup ac$, one has $C_1 \to_{S_1} C_2 \in ab$ or $C_1 \to_{S_2} C_2 \in ac$. Indeed, there should be the first time the path from a to b or c goes to $Com_b \cup Com_c$ in $C_1 \to_{S_3} C_2 \in ab \cup ac$ and the segment of the path before this point will witness $C_1 \to_{S_1} C_2 \in ab$ or $C_1 \to_{S_2} C_2 \in ac$.

Now, let's proceed to estimate the probabilities of these events. For $C_1 \in a|b|c$ we will have $C_1 \to_{S_1} C_2 \in a|c$, so

$$\mathbf{P}(C_1 \in a|b|c \text{ and } C_1 \to_{S_1} C_2 \in ab) \le \mathbf{P}(ab|c).$$

Similarly,

$$\mathbf{P}(C_1 \in a|b|c \text{ and } C_1 \to_{S_2} C_2 \in ac) \le \mathbf{P}(ac|b).$$

To estimate the probability of $\mathbf{P}\left(C_1 \in a|b|c \text{ and } C_1 \to_{S_3} C_2 \in (ab \cup ac)\right)$ we make use of the fact that if C_1 belongs to $a| \cap a|c$, then $\bar{S_3}$ contains a cut from a to b and c, so $C_1 \to_{\bar{S_3}} C_2$ also belongs to $a|b \cap a|c$.

$$\mathbf{P}\Big(C_1 \in a|b|c \text{ and } C_1 \to_{S_3} C_2 \in (ab \cup ac)\Big)$$
$$\ge \mathbf{P}\Big(C_1 \in (a|b \cap a|c) \text{ and } C_1 \to_{S_3} C_2 \in (ab \cup ac)\Big) - \mathbf{P}(a|bc)$$
$$= \mathbf{P}\Big(C_1 \to_{\bar{S_3}} C_2 \in (a|b \cap a|c) \text{ and } C_1 \to_{S_3} C_2 \in (ab \cup ac)\Big) - \mathbf{P}(a|bc) = \mathbf{P}(a|b \cap a|c)\mathbf{P}(ab \cup ac) - \mathbf{P}(a|bc)$$

Finally, this allows us to make the conclusion

$$\mathbf{P}(a|b \cap a|c)\mathbf{P}(ab \cup ac) \le \mathbf{P}(ab|c) + \mathbf{P}(ac|b) + \mathbf{P}(a|bc). \tag{4}$$

If $\mathbf{P}(abc) \ge p - \varepsilon$ and $\mathbf{P}(a|b|c) \ge 1 - p - \varepsilon$, this implies $(p-\varepsilon)(1-p-\varepsilon) \le 2\varepsilon$, which is false for small ε. □

For $p = \frac{1}{2}$, from the equation (4) one can conclude that $\mathbf{P}(abc)$ and $\mathbf{P}(a|b|c)$ can not be simultaneously greater than 0.37586. If we denote the maximal possible value of $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big)$ for any bond percolation by α_3, we get an estimate $\alpha_3 < 0.37586$, which we improve later in Appendix A.

5 Simulating 3-hyperedge: computer-assisted proof

Assume that we have a graph G with designated vertices a, b, c and a bond percolation on it. Let S_1, S_2 and S_3 be as in the previous section. Bond percolation induces a distribution ρ on $J = \{a|b|c, a|bc, ac|b, ab|c, abc\}$. In the same way, consider all possible 8-tuples

$$(C_1, C_2, C_1 \to_{S_1} C_2, C_1 \to_{\overline{S_1}} C_2, C_1 \to_{S_2} C_2, C_1 \to_{\overline{S_2}} C_2, C_1 \to_{S_3} C_2, C_1 \to_{\overline{S_3}} C_2)$$

and the probability distribution they induce on J^8. Some of the elements of J^8 are impossible for graph restrictions. We find these impossible elements by an algorithm[1].

Also, this probability distribution should have the same marginals $\rho \times \rho$ when restricted to pairs (C_1, C_2), $(C_1 \to_{S_1} C_2, C_1 \to_{\overline{S_1}} C_2)$, $(C_1 \to_{S_2} C_2, C_1 \to_{\overline{S_2}} C_2)$ and $(C_1 \to_{S_3} C_2, C_1 \to_{\overline{S_3}} C_2)$. All these restrictions produce a linear program, and one can see if it is feasible for different ρ's. One of the emerging restrictions on ρ is

$$\mathbf{P}(a|b \cap a|c)\mathbf{P}(ab \cup ac) \leq \mathbf{P}(ab|c) + \mathbf{P}(ac|b) + \mathbf{P}(a|bc) - 2\mathbf{P}(ab|c)\mathbf{P}(ac|b), \tag{5}$$

which is obviously better then inequality (4) and leads to an estimate $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big) \leq 0.369$. We double-checked manually that the tuples $(J_1, J_2, \ldots, J_8) \in J^8$ where dual potentials of the linear program add up to the negative number are indeed infeasible.

Moreover, surprisingly, it also proves inequality

$$\mathbf{P}(abc)\mathbf{P}(a|b|c) \geq \mathbf{P}(ab|c)\mathbf{P}(ac|b) + \mathbf{P}(ab|c)\mathbf{P}(a|bc) + \mathbf{P}(ac|b)\mathbf{P}(a|bc), \tag{6}$$

taken from the work [5], which generalizes the Harris–Kleitman inequality. Both of these computer-assisted proofs are available in the GitHub repository.

6 Further questions

Inequalitites (4) and (5) prove that if all three probabilities $\mathbf{P}(ab|c)$, $\mathbf{P}(ac|b)$ and $\mathbf{P}(a|bc)$ are 0, then one of $\mathbf{P}(abc)$ and $\mathbf{P}(a|b|c)$ should be 0. In fact, the stronger statement holds:

Proposition 6.1. *If* $\mathbf{P}(ab|c) = 0$*, then*

$$\mathbf{P}(a|b|c)\mathbf{P}(abc) = \mathbf{P}(ac|b)\mathbf{P}(a|bc).$$

Proof. As in the proof of Theorem 2.4, we first delete or contract certain edges. Now all paths from a to b should pass through c, otherwise there will be a nonzero probability of one such path being open

□

However, contrary to the inequalitites (4) and (5), this proof tells nothing when $\mathbf{P}(ab|c) < \varepsilon$. So, we pose two conjectures strengthening our results:

Conjecture 6.2. *For any* $\varepsilon > 0$ *there exists* $\delta > 0$ *such that if* $\mathbf{P}(ab|c) < \delta$ *and* $\mathbf{P}(ac|b) < \delta$*, then* $\mathbf{P}(abc)$ *or* $\mathbf{P}(a|b|c)$ *is less then* ε*.*

Conjecture 6.3. *For any* $\varepsilon > 0$ *there exists* $\delta > 0$ *such that if* $\mathbf{P}(ab|c) < \delta$*, then*

$$\mathbf{P}(abc)\mathbf{P}(a|b|c) - \mathbf{P}(ac|b)\mathbf{P}(a|bc) < \varepsilon.$$

It would also be interesting to find the exact value for α_3. The best boundaries are given in the Appendix A.

7 Acknowledgements

We thank Igor Pak for many helpful comments on the manuscript and Tom Hutchcroft for his thoughtful review and encouragement.

[1] https://github.com/Kroneckera/bunkbed
We loop through each element $(J_1, J_2, \ldots, J_8)$ of J^8. First, we split the vertices of G into "codes" based on which of the vertices a, b, c it lies together with in each of the configurations $(C_1, C_2, C_1 \to_{S_1} C_2, C_1 \to_{\overline{S_1}} C_2, C_1 \to_{S_2} C_2, C_1 \to_{\overline{S_2}} C_2, C_1 \to_{S_3} C_2, C_1 \to_{\overline{S_3}} C_2)$. Then we build "universal" graphs $\vec{G}_1$ and $\vec{G}_2$, using these codes as vertices, including all edges except for the edges between codes in different parts of the graph. Finally, we use $\vec{G}_1$ and $\vec{G}_2$ as C_1 and C_2, construct the remaining elements of the 8-tuple and use graph search algorithms to check whether they coincide with $(J_1, J_2, \ldots, J_8)$.

References

[1] Jacob van den Berg and Harry Kesten. Inequalities with applications to percolation and reliability. *Journal of applied probability* **22.3** (1985): 556-569.

[2] Jacob van den Berg, Olle Häggström, and Jeff Kahn. Some conditional correlation inequalities for percolation and related processes. *Random Structures & Algorithms* **29.4** (2006): 417-435.

[3] Michael E. Fisher. Critical probabilities for cluster size and percolation problems. *Journal of Mathematical Physics* **2.4** (1961): 620-627.

[4] Michael E. Fisher and John W. Essam. Some cluster size and percolation problems. *Journal of Mathematical Physics* **2.4** (1961): 609-619.

[5] Nikita Gladkov. A strong FKG inequality for multiple events, arXiv:2305.02653, 2023, 7 pp.

[6] Geoffrey Grimmett. *Probability on graphs: random processes on graphs and lattices.* Vol. **8**. Cambridge University Press, 2018.

[7] Tom Hutchcroft. Power-law bounds for critical long-range percolation below the upper-critical dimension. *Probability theory and related fields* **181** (2021): 533-570.

[8] Adam Zsolt Wagner. Constructions in combinatorics via neural networks, arXiv:2104.14516, 2021, 23 pp.

[9] John C. Wierman and Robert M. Ziff. Self-dual planar hypergraphs and exact bond percolation thresholds. *The Electronic Journal of Combinatorics* **18.1** (2011) 19 pp.

A Appendix: optimizing α_3

Let us recall that α_3 denotes the largest possible value of $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big)$ for the bond percolation. Let us restrict ourselves to the triangle graph with all three probabilities equal to p. Then $\mathbf{P}(a|b|c) = (1-p)^3$ and $\mathbf{P}(abc) = p^3 + 3p^2(1-p)$. These numbers coinside for $p \approx 0.3473$, and we get $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big) \approx 0.278$, a root of the equation $x^3 - 24x^2 + 3x + 1 = 0$.

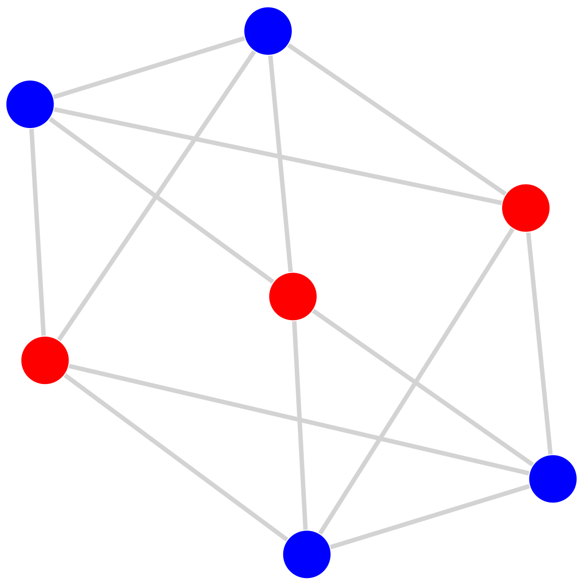

Figure 3: Graph for α_3.

One can do better by utilizing the graph in Figure 3 where each red-blue edge has probability 0.32537 and both blue-blue edges have probability 0.19231. This way we get $\mathbf{P}(abc) \approx \mathbf{P}(a|b|c) \approx 0.29065$.

Our computer search using algorithms from Wagner [8] wasn't able to beat this estimate (See the best $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big)$ achieved on each training epoch in Figure 4).

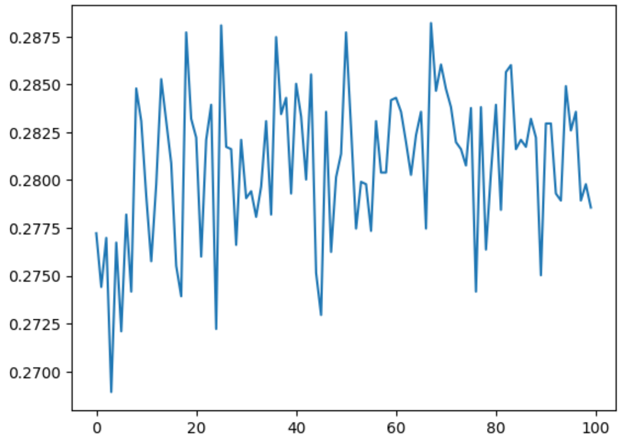

Figure 4: Best $\min\big(\mathbf{P}(abc), \mathbf{P}(a|b|c)\big)$ achieved on each training epoch.

In fact, if $\mathbf{P}(abc) = \mathbf{P}(a|b|c)$, it seems this probability can only lie in a narrow range from 0.27 to 0.291. Indeed, in this case inequality (6) gives the lower bound of $2 - \sqrt{3} \approx 0.2679$.