Approximating Cumulative Pebbling Cost Is Unique Games Hard

Jeremiah Blocki

Department of Computer Science, Purdue University, West Lafayette, IN, USA
https://www.cs.purdue.edu/homes/jblocki
jblocki@purdue.edu

Seunghoon Lee

Department of Computer Science, Purdue University, West Lafayette, IN, USA
https://www.cs.purdue.edu/homes/lee2856
lee2856@purdue.edu

Samson Zhou

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
https://samsonzhou.github.io/
samsonzhou@gmail.com

—— **Abstract** ——

The cumulative pebbling complexity of a directed acyclic graph G is defined as $\mathsf{cc}(G) = \min_P \sum_i |P_i|$, where the minimum is taken over all legal (parallel) black pebblings of G and $|P_i|$ denotes the number of pebbles on the graph during round i. Intuitively, $\mathsf{cc}(G)$ captures the amortized Space-Time complexity of pebbling m copies of G in parallel. The cumulative pebbling complexity of a graph G is of particular interest in the field of cryptography as $\mathsf{cc}(G)$ is tightly related to the amortized Area-Time complexity of the Data-Independent Memory-Hard Function (iMHF) $f_{G,H}$ [7] defined using a constant indegree directed acyclic graph (DAG) G and a random oracle $H(\cdot)$. A secure iMHF should have amortized Space-Time complexity as high as possible, e.g., to deter brute-force password attacker who wants to find x such that $f_{G,H}(x) = h$. Thus, to analyze the (in)security of a candidate iMHF $f_{G,H}$, it is crucial to estimate the value $\mathsf{cc}(G)$ but currently, upper and lower bounds for leading iMHF candidates differ by several orders of magnitude. Blocki and Zhou recently showed that it is NP-Hard to compute $\mathsf{cc}(G)$, but their techniques do not even rule out an efficient $(1 + \varepsilon)$-approximation algorithm for any constant $\varepsilon > 0$. We show that for *any* constant $c > 0$, it is Unique Games hard to approximate $\mathsf{cc}(G)$ to within a factor of c.

Along the way, we show the hardness of approximation of the DAG Vertex Deletion problem on DAGs of constant indegree. Namely, we show that for any $k, \varepsilon > 0$ and given a DAG G with N nodes and constant indegree, it is Unique Games hard to distinguish between the case that G is (e_1, d_1)-reducible with $e_1 = N^{1/(1+2\varepsilon)}/k$ and $d_1 = kN^{2\varepsilon/(1+2\varepsilon)}$, and the case that G is (e_2, d_2)-depth-robust with $e_2 = (1 - \varepsilon)ke_1$ and $d_2 = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$, which may be of independent interest. Our result generalizes a result of Svensson who proved an analogous result for DAGs with indegree $\mathcal{O}(N)$.

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).
Editor: Thomas Vidick; Article No. 13; pp. 13:1–13:27

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1   Introduction

The black pebbling game is a powerful abstraction that allows us to analyze the complexity of functions f_G with a static data-dependency graph G. In particular, a directed acyclic graph (DAG) $G = (V, E)$ can be used to encode data-dependencies between intermediate values produced during computation e.g., if L_v is the vth intermediate value and $L_v := L_j \times L_i$ then the DAG G would include directed edges (i, v) and (j, v) indicating that L_v depends on the previously computed values L_i and L_j. A black pebbling of G is a sequence $P = (P_0, \dots, P_t) \subseteq V$ of pebbling configurations. Intuitively, a pebbling configuration P_i describes the set of data labels that have been computed and stored in memory at time i. The rules of the pebbling game stipulate that we must have $\mathsf{parents}(v) = \{u \ : \ (u, v) \in E\} \subseteq P_i$ for each newly pebbled node $v \in P_{i+1} \setminus P_i$ i.e., before we can compute a new data value L_v, we must first have the labels of each dependent data value L_u available in memory.

Historically, much of the literature has focused on the sequential black pebbling game where we require that $|P_{i+1} \setminus P_i| \leq 1$ for all round i. In recent years, the parallel black pebbling game has seen renewed interest due to the rapid expansion of parallel computing, e.g., GPUs, FPGAs. In the more general parallel black pebbling game, there is no such restriction on the number of new pebbles in each round, i.e., on a parallel architecture, it is possible to determine L_v for each node $v \in P_{i+1} \setminus P_i$ simultaneously since the dependent data-values are already in memory.

There are several natural ways to measure the cost of a pebbling. The space complexity of a DAG G asks for a legal pebbling $P = (P_0, \dots, P_t)$ that minimizes the maximum space usage $\max_{i \leq t} |P_i|$ – even if the time t is exponential in the number of nodes N. Space-time complexity asks for a legal pebbling $P = (P_0, \dots, P_t)$ that minimizes the space-time product $t \times \max_{i \leq t} |P_i|$. Alwen and Serbinenko [7] observed that in the parallel black pebbling game, the space-time of pebbling $G^{\times m}$, m independent copies of a DAG G, does not always scale linearly with m. In particular, for some DAGs G the total space-time cost of pebbling $G^{\times m}$ is roughly equal to the space-time cost of pebbling a single instance of G for $m = \tilde{\mathcal{O}}(\sqrt{N})$!

Alwen and Serbinenko [7] introduced the notion of the cumulative pebbling cost $\mathsf{cc}(G)$ of a DAG G to model the amortized space-time costs in the parallel black pebbling game. Formally, the cumulative pebbling cost of a pebbling P is given by $\mathsf{cc}(P) = \sum_i |P_i|$ and $\mathsf{cc}(G) = \min_P \mathsf{cc}(P)$, where the minimum is taken over all legal (parallel) black pebblings of G. The cumulative pebbling cost is a fundamental metric that is worth studying. It captures the amortized space-time cost of pebbling m copies of G in parallel, i.e., in the limit we have $\mathsf{cc}(G) = \lim_{m \to \infty} \mathsf{ST}(G^{\times m})/m$ where the space-time cost of a pebbling $P = (P_1, \dots, P_t)$ is $\mathsf{ST}(P) = t \times \max_i |P_i|$ and the notation $G^{\times m}$ denotes a new graph consisting of m disjoint copies of G.

In this paper, we address the following question:

> *Given a DAG G, can we (approximately) compute $\mathsf{cc}(G)$?*

This is a natural question in settings where we want to evaluate the function f_G (with data-dependency DAG G) on many distinct inputs – $\mathsf{cc}(G)$ models the amortized cost of computing f_G. The question is also highly relevant to the cryptanalysis of Data-Independent Memory-Hard Functions (iMHFs). In the context of password hashing we want to find a (constant indegree) DAG G with maximum cumulative pebbling complexity, e.g., to maximize

the cost of a brute-force attacker who wants to evaluate the function f_G on every input in a password cracking dictionary. Thus, given a DAG G one might wish to lower-bound $\mathsf{cc}(G)$ before using G in the design of a memory-hard password hashing algorithm.

Cumulative Pebbling Complexity in Cryptography

In many natural contexts such as password hashing and Proofs of Work, it is desirable to lower bound the amortized space-time cost, e.g., in the random oracle model it is known that the cumulative memory complexity of a (side-channel resistant) iMHF $f_{G,H}$ is $\Omega(\mathsf{cc}(G))$, where $f_{G,H}$ is a labeling function defined in terms of the DAG G and a random oracle H [7]. Thus, in the field of cryptography there has been a lot of interest in designing constant indegree graphs with cumulative pebbling cost $\mathsf{cc}(G)$ as large as possible and in analyzing the pebbling cost $\mathsf{cc}(G)$ of candidate iMHF constructions $f_{G,H}$, e.g., see [2, 5, 3, 6, 4, 14].

From an asymptotic standpoint many of the open questions have been (nearly) resolved. Alwen and Blocki [2] showed that for any DAG G with N nodes and constant indegree we have $\mathsf{cc}(G) = \mathcal{O}\left(N^2 \log \log N / \log N\right)$, while Alwen et al. [5, 4] gave constructions with $\mathsf{cc}(G) = \Omega(N^2/\log N)$. For Argon2i, the winner of the password hashing competition, we have the upper bound $\mathsf{cc}(G) = \mathcal{O}\left(N^{1.767}\right)$ and the lower bound $\mathsf{cc}(G) = \tilde{\Omega}\left(N^{1.75}\right)$ [14].

Most of these upper/lower bounds exploited a relationship between $\mathsf{cc}(G)$ and a combinatorial property called depth-robustness. A DAG $G = (V, E)$ is (e, d)-reducible if we can find a subset $S \subseteq V$ with $|S| \leq e$ such that any directed path P in G of length d contains at least one node in S. On the other hand, if G is not (e, d)-reducible, then we say that G is (e, d)-depth robust. Depth-robustness is known to be both necessary [2] and sufficient [5] for secure iMHFs. In particular, any (e, d)-reducible DAG G with N nodes and indegree $\mathsf{indeg}(G)$ has $\mathsf{cc}(G) \leq \min_{g \geq d}\left(eN + gN \times \mathsf{indeg}(G) + \frac{N^2 d}{g}\right)$ [2] while any (e, d)-depth robust DAG G has $\mathsf{cc}(G) \geq ed$ [5]. The later observation was used to build a constant indegree graph G with $\mathsf{cc}(G) = \Omega(N^2/\log N)$ by showing that the constructed G is $(\Omega(N/\log N), \Omega(N))$-depth robust. The former observation was used to prove that any constant indegree graph has $\mathsf{cc}(G) = \mathcal{O}\left(N^2 \log \log N / \log N\right)$ by exploiting the observation that any such DAG G is $\left(\mathcal{O}\left(N \log \log N / \log N\right), \Omega(N/\log^2 N)\right)$-reducible (simply set $g = \mathcal{O}\left(N \log \log N / \log N\right)$ in the above [2] bound).

Although many of the open questions have been (nearly) resolved from an asymptotic standpoint, from a concrete security standpoint for all practical iMHF candidates G, the best known upper and lower bounds on $\mathsf{cc}(G)$ differ by several orders of magnitude. In fact, Blocki et al. [11] recently found that for practical parameter settings ($N \leq 2^{24}$), Argon2i provides better resistance to known pebbling attacks than DRSample [4] despite the fact that DRSample ($\mathsf{cc}(G) = \Omega(N^2/\log N)$) is asymptotically superior to Argon2i ($\mathsf{cc}(G) = \tilde{\Omega}\left(N^{1.75}\right)$). Of course it is certainly possible that an improved pebbling strategy for Argon2i will reverse this finding tomorrow making it difficult to provide definitive recommendations about which construction is superior in practice.

Given a DAG G, one might try to resolve these questions directly by (approximately) computing $\mathsf{cc}(G)$. Blocki and Zhou [15] previously showed that the problem of computing $\mathsf{cc}(G)$ is NP-Hard. However, their result does not even rule out the existence of a $(1 + \varepsilon)$-approximation algorithm for any constant $\varepsilon > 0$.

1.1   Our Contributions

Our main result is the hardness of any constant factor approximation to the cost of graph pebbling even for DAGs with constant indegree[1].

▶ **Theorem 1.** *Given a DAG G with constant indegree, it is Unique Games hard to approximate $\mathsf{cc}(G)$ within any constant factor. (See Theorem 13.)*

Along the way to proving our main result, we show that for any constant $k > 0, \varepsilon > 0$, given a constant indegree graph G, it is Unique Games hard to distinguish between the following two cases: (1) G is (e_1, d_1)-reducible with $e_1 = N^{1/(1+2\varepsilon)}/k$ and $d_1 = kN^{2\varepsilon/(1+2\varepsilon)}$ and (2) G is (e_2, d_2)-depth-robust with $e_2 = (1 - \varepsilon)ke_1$ and $d_2 = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$. This intermediate result (see Corollary 8) generalizes a result of Svensson [45], who proved an analogous result for DAGs G with arbitrarily large indegree $\mathsf{indeg}(G) = \mathcal{O}(N)$.

Corollary 8 may be of independent interest as depth-robust graphs have found many other applications in cryptography including Proofs of Sequential Work [37], Proofs of Space [24], Proofs of Replication [39, 25] and (relaxed) locally correctable codes for computationally bounded channels [10, 12]. Testing the depth-robustness of a DAG G is especially relevant to the analysis of (tight) Proofs of Space/Replication – several constructions rely on (unproven) conjectures about the concrete depth-robustness of particular DAGs e.g., see [16, 25].

1.2   Technical Ingredients

To prove our result we use three technical ingredients. The first ingredient is a reduction of Svensson [45] that it is Unique Games hard to distinguish between a DAG G (with $\mathsf{indeg}(G) = \mathcal{O}(N)$) that is (e_1, d_1)-reducible or (e_2, d_2)-depth-robust. The second technical ingredient is γ-Extreme Depth-Robust Graphs [6] with bounded indegree. We use γ-Extreme Depth-Robust Graphs to modify the construction of Svensson [45] and show that the same result holds for graphs with much smaller indegree. Finally, we use low depth superconcentrators to boost the lower bound on cc to $\min\{e_2N, d_2N\}/8$ instead of e_2d_2 in the case the graph is (e_2, d_2)-depth robust. We prove that this can be done without significantly increasing the pebbling cost in the case the graph is (e_1, d_1)-reducible.

1.2.1   Technical Ingredient 1

Our first technical ingredient is a result of Svensson [45], who proved that for any constant $k > 0, \varepsilon > 0$, it is Unique Games hard to distinguish between the following two cases (1) G is (e_1, d_1)-reducible with $e_1 = N/k$ and $d_1 = k$, or (2) G is (e_2, d_2)-depth robust with $e_2 = N(1-1/k)$ and $d_2 = \Omega(N^{1-\varepsilon})$. To prove this, Svensson gave a reduction that transforms from any instance of Unique Games $\mathcal{U}$ to a directed acyclic graph $G_\mathcal{U}$ on N nodes such that $G_\mathcal{U}$ is (e_1, d_1)-reducible for $e_1 \approx N/k$ and $d = k$ if $\mathcal{U}$ is satisfiable. Otherwise, if $\mathcal{U}$ is unsatisfiable, it can be shown that $G_\mathcal{U}$ is (e_2, d_2)-depth robust. This is a potentially useful starting point because the pebbling complexity of a graph $G_\mathcal{U}$ is closely related to its depth-robustness. In particular, in the second case, a result of Alwen et al. [5] establishes that $\mathsf{cc}(G_\mathcal{U}) \geq e_2d_2$ and in the first case, a result of Alwen and Blocki shows that $\mathsf{cc}(G_\mathcal{U}) \leq \min_{g \geq d_1}\left(e_1N + gN \times \mathsf{indeg}(G_\mathcal{U}) + \frac{N^2d_1}{g}\right)$ [2].

---

[1] Each node v in a data-dependency DAG G model an atomic unit of computation. Thus, in practice we expect G to have indegree 2 or 3. If $L_v = g(L_{v_1}, \ldots, L_{v_k})$ is a function of $k \gg 2$ previously computed values $L_{v_1}, \ldots, L_{v_k}$ then we would have generated several additional intermediate data-values while evaluating $g(\cdot)$. These data-values should have been included as nodes in G which is supposed to have a node for every intermediate data-value.

Challenges of Applying Svensson's Construction

While the pebbling complexity of $G_{\mathcal{U}}$ is related to depth-robustness, there is still a vast gap between the upper/lower bounds. In particular, in Svensson's construction we have $\mathsf{indeg}(G_{\mathcal{U}}) = \mathcal{O}(N)$, so the $gN \times \mathsf{indeg}(G_{\mathcal{U}})$ term could be as large as $gN^2 \gg e_2 d_2$. Thus, we would need to be able to reduce the indegree significantly to obtain a gap between $\mathsf{cc}(G_{\mathcal{U}})$ in the two cases. (In fact, we can show the that pebbling cost is exactly $\mathsf{cc}(G_{\mathcal{U}}) = \frac{N(L+1)}{2}$ independent of the Unique Games instance $\mathcal{U}$ – see Lemma 17 in the appendix.) We remark that a naïve attempt to reduce indegree in Svensson's construction $G_{\mathcal{U}}$ by replacing every node v (as in [5]) with a path of length $N + \mathsf{indeg}(v)$ would result in a constant indegree graph $G'_{\mathcal{U}}$ with $N' \approx 2N^2$ nodes that will not be useful for our purposes. The new graph $G'_{\mathcal{U}}$ would be (e_1, d_1)-reducible in the first case with $e_1 = N/k = \mathcal{O}\left(\sqrt{N'}/k\right)$ and $d_1 = 2kN = \mathcal{O}\left(\sqrt{N'}k\right)$. In the second case, the DAG $G'_{\mathcal{U}}$ would be (e_2, d_2)-depth robust with $e_2 \approx ke_1$ and $d_2 = \mathcal{O}\left(N'^{1-\varepsilon/2}\right)$. We would now have $\mathsf{cc}(G'_{\mathcal{U}}) \leq \min_{g \geq d_1}\left(e_1 N' + 2gN' + \frac{N'^2 d_1}{g}\right) = \omega(e_1 N')$ for our upper bound while the lower bound is at most $e_2 d_2 \approx ke_1 N'^{1-\varepsilon/2}$. At the end of the day, the graph $G_{\mathcal{U}}$ is still quite far from what we need.

1.2.2 Technical Ingredient 2: γ-Extreme Depth-Robust Graphs

It does not seem to be possible to obtain a suitable graph $G_{\mathcal{U}}$ by applying indegree reduction techniques to Svensson's Construction in a black-box manner. Instead, we open up the black-box and show how to reduce the indegree using a recent technical result of Alwen et al. [6]. A DAG $G_{\gamma,N}$ on N nodes is said to be γ-extreme depth-robust if it is (e, d)-depth robust for any $e, d > 0$ such that $e + d \leq (1 - \gamma)N$. Alwen et al. [6] showed that for any constant $\gamma > 0$, there exists a family $\{G_{\gamma,N}\}_{N=1}^{\infty}$ of γ-extreme depth robust DAGs with maximum indegree $\mathcal{O}(\log N)$. While Alwen et al. [6] were not focused on outdegree, it is not too difficult to see that their construction yields a single family of DAGs with maximum indegree and outdegree $\mathcal{O}(\log N)$.

In Svensson's construction, the DAG $G_{\mathcal{U}}$ is partitioned into $L = N^{1-\varepsilon}$ symmetric layers i.e., if u_ℓ (the copy of node u in layer ℓ_1) is connected to v_{ℓ_2} (the copy of node v in layer $\ell_2 > \ell_1$) then for *any* layers $i < j \leq L$, the directed edge (u_i, v_j) exists. The fact that this edge is "copied" $\mathcal{O}(L^2)$ times for every pair of layers $i < j$ significantly increases the indegree. However, Svensson's argument that $G_{\mathcal{U}}$ is depth-robust in the second case relies on the existence of each of these edges. To reduce the indegree we start with a γ-extreme depth robust DAG $G_{\gamma,L}$ on L nodes and only keep edges between nodes u_i and v_j in layers i and j if there is a path of length ≤ 2 between nodes i and j in $G_{\gamma,L}$. The new graph can also be shown to have degree at most $\mathcal{O}\left(\mathsf{indeg}(G_L) \times \mathsf{outdeg}(G_L) \times N/L\right) = \mathcal{O}\left(N^\varepsilon \log^2 N\right)$. Despite the fact that the indegree is vastly reduced, we are still able to modify Svensson's argument to prove that (for a suitable constant $\gamma > 0$) our new graph is still (e_2, d_2)-depth robust with $e_2 \approx ke_1$ and $d_2 = \mathcal{O}\left(N^{1-\varepsilon}\right)$ – note that the new graph is clearly still (e_1, d_1)-reducible if $\mathcal{U}$ is satisfiable since we only remove edges from Svensson's construction.

We can then apply the generic black-box indegree reduction of [5] to reduce the indegree to 2 by replacing every node with a path of length $N^{2\varepsilon}$. This established our first technical result that even for constant indegree DAGs, it is Unique Games hard to distinguish between the following two cases: (1) G is (e_1, d_1)-reducible with $e_1 = N^{1/(1+2\varepsilon)}/k$ and $d_1 = kN^{2\varepsilon/(1+2\varepsilon)}$, and (2) G is (e_2, d_2)-depth-robust with $e_2 = (1 - \varepsilon)ke_1$ and $d_2 = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$.

1.2.3   Technical Ingredient 3: Superconcentrators

Although indegree reduction is a crucial step toward showing hardness of approximation for graph pebbling complexity, we still cannot apply known results that relate (e_1, d_1)-reducibility and (e_2, d_2)-depth robustness to pebbling complexity, since there is still no gap between the pebbling complexity of the two cases. In particular, we are always stuck with the $e_1 N$ term in the upper bound of [2] which is *already* much larger than the lower bound $e_2 d_2$ from [6]. To overcome this result we rely on superconcentrators. A *superconcentrator* is a graph that connects N input nodes to N output nodes so that any subset of k inputs and k outputs are connected by k vertex disjoint paths. Moreover, the total number of edges in the graph should be $\mathcal{O}(N)$.

Blocki et al. [11] recently proved that G', the *superconcentrator overlay* of an (e, d)-depth robust graph, has pebbling cost $\mathsf{cc}(G') \geq \max\{eN, dN\}/8$, which is a significant improvement on the lower bound $\mathsf{cc}(G') \geq ed$ when $e = o(N)$ and $d = o(N)$. This allows us to increase the lower-bound in case 2, but we need to be careful that we do not significantly increase the pebbling cost in case 1. To do this we rely on the existence of superconcentrators with depth $\mathcal{O}(\log N)$ [40] and we give a significantly improved pebbling attack on the superconcentrator overlay DAG G' in case 1 when the original graph is (e_1, d_1)-reducible. With the improved pebbling attack, we are able to show that $\mathsf{cc}(G) \geq e_1 k N/16$ in case 2 and that $\mathsf{cc}(G) \leq 16 e_1 N$ in case 1. Since k is an arbitrary constant, this implies that it is Unique Games hard to approximate $\mathsf{cc}(G)$ to within any constant factor $c > 0$.

2   Related Work

Pebbling games have found a number of applications under various formulations and models (see the survey [38] for a more thorough review). The sequential black pebbling game was introduced by Hewitt and Paterson [29], and by Cook [19] and has been particularly useful in exploring space/time trade-offs for various problems like matrix multiplication [47], fast fourier transformations [43, 47], integer multiplication [46] and many others [17, 44]. In cryptography it has been used to construct/analyze Proofs of Space [24, 41], Proofs of Work [23, 37] and Memory-Hard Functions [26]. Alwen and Serbinenko [7] argued that the parallel version of the black pebbling game was more appropriate for Memory-Hard Functions and they proved that any iMHF attacker in the parallel random oracle model corresponds to a pebbling strategy with equivalent cumulative memory cost.

The space cost of the black pebbling game is defined to be $\max_i |P_i|$, which intuitively corresponds to minimizing the maximum space required during computation of the associated function. Gilbert et al. [27] studied the space-complexity of the black-pebbling game and showed that this problem is PSPACE-Complete by reducing from the truly quantified boolean formula (TQBF) problem. In our case, the decision problem is $\mathsf{cc}(G) \leq k$ is in NP because the optimal pebbling strategy cannot last for more than N^2 steps since *any* graph with N nodes has $\mathsf{cc}(G) \leq N^2$.

Red-Blue Pebbling

Given a DAG $G = (V, E)$, the goal of the red-blue pebbling game [30] is to place pebbles on all sink nodes of G (not necessarily simultaneously) from an empty starting configuration. Intuitively, red pebbles represent values in cache and blue pebbles represent values stored in memory. Blue pebbles must be converted to red pebbles (e.g., loaded into cache) before

they can be used in computation, but there is a limit m (cache-size) on the number of red-pebbles that can be used. Red-blue pebbling games have been used to study memory-bound functions [22] (functions that incur many expensive cache-misses [1]).

Ren and Devadas introduced the notion of bandwidth hard functions and used the red-blue pebbling game to analyze the energy cost of a memory hard function [42]. In their model, red-moves (representing computation performed using data in cache) have a smaller cost c_r than blue-moves c_b (representing data movements to/from memory) and a DAG G on N nodes is said to be bandwidth hard if any red-blue pebbling has cost $\Omega(N \cdot c_b)$. Ren and Devadas showed that the bit reversal graph [35], which forms the core of iMHF candidate Catena-BRG [26], is maximally bandwidth hard. Subsequently, Blocki et al. [13] gave a pebbling reduction showing that any attacker random oracle model (pROM) can indeed be viewed as a red-blue pebbling with equivalent cost. They also show that it is NP-Hard to compute the minimum cost red-blue pebbling of a DAG G i.e., the decision problem "is the red-blue pebbling cost $\leq k$?" is NP-Complete (A result of Demaine and Liu [20, 36] implies that the problem is PSPACE-Hard to compute the red-blue pebbling cost when $c_r = 0$ i.e., computation is free). In general, the red-blue cost of G is always lower bounded by $c_r N$ and upper-bounded by $2c_b N + c_r N$. The question of a more efficient c-approximation algorithm for $c = o(c_b/c_r)$ remains open.

Unique Games

Recently, the Unique Games Conjecture and related conjectures have received a lot of attention for their applications in proving hardness of approximation. Khot et al. [32] showed that the Goemans-Williamson approximation algorithm for Max-Cut [28] is optimal, assuming the Unique Games Conjecture. Khot and Regev [34] showed that Minimum Vertex Cover problem is Unique Games hard to solve within a factor of $2 - \varepsilon$, which is nearly tight from the guarantee that a simple greedy algorithm gives. The Unique Games Conjecture also leads to tighter approximation hardness for other problems including Max 2-SAT [32] and Betweenness [18]. Although a previous stronger version of the conjecture asked whether Unique Games instances required exponential time algorithms in the worst case, Arora et al. [8] gave a subexponential time algorithm for Unique Games. Lately, focus has also been drawn toward studying the related Label Cover Problem, such as the 2-Prover-1-Round Games, i.e. the 2-to-1 Games Conjecture [21] and the 2-to-2 Games Conjecture [33].

3 Preliminaries

We use the notation $[N]$ to denote the set $\{0, 1, \ldots, N - 1\}$. Given a directed acyclic graph $G = (V, E)$ and a node $v \in V$, we use $\mathsf{parents}(v) = \{u : (u, v) \in E\}$ (resp. $\mathsf{children}(v) = \{u : (v, u) \in E\}$) to denote the parents (resp. children) of node v. We use $\mathsf{indeg}(v) = |\mathsf{parents}(v)|$ (resp. $\mathsf{outdeg}(v) = |\mathsf{children}(v)|$) to denote the number of incoming (resp. outgoing) edges into (resp. out of) the vertex v. We also define $\mathsf{indeg}(G) = \max_{v \in V} \mathsf{indeg}(v)$ and $\mathsf{outdeg}(G) = \max_{v \in V} \mathsf{outdeg}(v)$. Given a set $S \subseteq V$ of nodes, we use $G - S$ to refer to the graph obtained by deleting all nodes in S and all edges incident to S. We also use $G[S] = G - (V \setminus S)$ to refer to the subgraph induced by the nodes S, i.e., deleting every other node in $V \setminus S$. Given a node $v \notin S$, we use $\mathsf{depth}(v, G - S)$ to refer to the longest directed path in $G - S$ ending at node v and we use $\mathsf{depth}(G - S) = \max_{v \notin S} \mathsf{depth}(v, G - S)$ to refer to the longest directed path in $G - S$. Given a subset B, we will also use $\mathsf{depth}_B(v, G - S)$ to refer to the maximum number of nodes in the set B contained in any directed path in $G - S$ that ends at node v. We define $\mathsf{depth}_B(G - S) = \max_{v \notin S} \mathsf{depth}_B(v, G - S)$ analogously.

▶ **Definition 2** (Unique Games). *An instance $\mathcal{U} = (G = (V, W, E), [R], \{\pi_{v,w}\}_{v,w})$ of Unique Games consists of a regular bipartite graph $G(V, W, E)$ and a set $[R]$ of labels. Each edge $(v, w) \in E$ has a constraint given by a permutation $\pi_{v,w} : [R] \to [R]$. The goal is to output a labeling $\rho : (V \cup W) \to [R]$ that maximizes the number of satisfied edges, where an edge is satisfied if $\rho(v) = \pi_{v,w}(\rho(w))$.*

▶ **Conjecture 3** (Unique Games Conjecture, [31]). *For any constants $\alpha, \beta > 0$, there exists a sufficiently large integer R (as a function of α, β) such that for Unique Games instances with label set $[R]$, no polynomial time algorithm can distinguish whether: (1) the maximum fraction of satisfied edges of any labeling is at least $1 - \alpha$, or (2) the maximum fraction of satisfied edges of any labeling is less than β.*

Graph Pebbling

The goal of the (black) pebbling game is to place pebbles on all sink nodes of some input directed acyclic graph (DAG) $G = (V, E)$. The game proceeds in rounds, and each round i consists of a number of pebbles $P_i \subseteq V$ placed on a subset of the vertices. Initially, the graph is unpebbled, $P_0 = \emptyset$, and in each round $i \geq 1$, we may place a pebble on $v \in P_i$ if either all parents of v contained pebbles in the previous round ($\mathsf{parents}(v) \subseteq P_{i-1}$) or if v already contained a pebble in the previous round ($v \in P_{i-1}$). In the sequential pebbling game, at most one new pebble can be placed on the graph in any round (i.e., $|P_i \backslash P_{i-1}| \leq 1$), but this restriction does not apply in the parallel pebbling game.

We use $\mathcal{P}_G^{\parallel}$ to denote the set of all valid parallel pebblings of G. The *cumulative cost* of a pebbling $P = (P_1, \ldots, P_t) \in \mathcal{P}_G^{\parallel}$ is the quantity $\mathsf{cc}(P) := |P_1| + \ldots + |P_t|$ that represents the sum of the number of pebbles on the graph during every round. The (parallel) *cumulative pebbling cost* of G, denoted $\mathsf{cc}(G) := \min_{P \in \mathcal{P}_G^{\parallel}} \mathsf{cc}(P)$, is the cumulative cost of the best legal pebbling of G.

A DAG G is (e, d)-*reducible* if there exists a subset $S \subseteq V$ of size $|S| \leq e$ such that $\mathsf{depth}(G - S) < d$. That is, there are no directed paths containing d vertices remaining, once the vertices in the set S are removed from G. If G is not (e, d)-reducible, we say that it is (e, d)-depth robust.

4 Reduction

Svensson [45] showed that for any constant $k, \epsilon > 0$ it is Unique Games hard to distinguish between whether a DAG G is (e_1, d_1)-reducible for $e_1 = N/k$ and $d_1 = k$ or G is (e_2, d_2)-depth robust with $e_2 = N(1 - 1/k)$ and $d_2 = \Omega(N^{1-\varepsilon})$. To prove this, Svensson showed how to transform a Unique Games instance $\mathcal{U} = (G = (V, W, E), [R], \{\pi_{v,w}\}_{v,w})$ into a graph $G_{\mathcal{U}}$ such that $G_{\mathcal{U}}$ is (e_1, d_1)-reducible if it is possible to satisfy $1 - \alpha$ fraction of the edges and $G_{\mathcal{U}}$ is (e_2, d_2)-depth robust if it is not possible to satisfy β-fraction of the edges. To obtain inapproximability results for cc, it is crucial to substantially reduce the indegree of this construction.

4.1 Review of Svensson's Construction

To construct $G_{\mathcal{U}}$, Svensson first constructs a layered bipartite DAG $\hat{G}_{\mathcal{U}}$, which encodes the unique games instance $\mathcal{U}$ and later transforms $\hat{G}_{\mathcal{U}}$ into the required DAG $G_{\mathcal{U}}$. For completeness, we provide a full description of the DAG $\hat{G}_{\mathcal{U}}$ in the appendix. We will focus our discussion here on the essential properties of the DAG $\hat{G}_{\mathcal{U}}$.

The graph $\hat{G}_\mathcal{U}$ has a number of *bit-vertices* B partitioned into *bit-layers* $B = B_0 \cup \ldots \cup B_L$, where B_i is the set of bit-vertices in bit-layer i. Each B_i can be partitioned into sets $B_{i,w}$ for $w \in W$. Similarly, $\hat{G}_\mathcal{U}$ has a number of *test-vertices* T partitioned into *test-layers* $T = T_0 \cup \ldots \cup T_{L-1}$, where T_i is the set of test-vertices in test-layer i. Outgoing edges for test-layer T_ℓ must be directed into a bit vertex in layer $B_{\ell'}$ with $\ell' > \ell$. Similarly, outgoing edges from B_ℓ must be directed into a test vertex in layer $T_{\ell'}$ with $\ell' \geq \ell$. Each T_i can be partitioned into sets $T_{i,v}$ for $v \in V$. The constraints in our Unique Games instance $\mathcal{U}$ are encoded as edges between the bit vertices and test vertices. We use $N = |T|$ to denote the total number of test nodes and remark that the parameter L is set such that $L \geq N^{1-\epsilon}$.

$\hat{G}_\mathcal{U}$ also displays symmetry between the layers in the sense that $B_\ell = \{b_1^\ell, \ldots, b_m^\ell\}$ and $T_\ell = \{t_1^\ell, \ldots, t_p^\ell\}$, so that the number of bit-vertices in each bit-layer is the same and the number of test-vertices in each test-layer is the same.

Symmetry

In Svensson's construction, we have exactly m bit vertices in every layer $B_\ell = \{b_1^\ell, \ldots, b_m^\ell\}$ and exactly p test vertices in every layer $T_\ell = \{t_1^\ell, \ldots, t_p^\ell\}$. The edges between B_ℓ and T_ℓ (resp. T_ℓ and $B_{\ell+1}$) encode the edge constraints in the unique games instance $\mathcal{U}$. Furthermore, the construction is symmetric so that directed edge (b_i^ℓ, t_j^ℓ) exists if and only if for *every* $\ell' \geq \ell$ the edge $(b_i^\ell, t_j^{\ell'})$ exists. Thus for any $\ell' \geq \ell$, the edges between B_ℓ and $T_{\ell'}$ encode the constraints in $\mathcal{U}$. Similarly, the directed edge $(t_j^\ell, b_i^{\ell+1})$ exists if and only if any $\ell' > \ell$ the edge $(t_j^\ell, b_i^{\ell'})$ exists. We remark that this means that the indegree of the graph $\hat{G}_\mathcal{U}$ is at least L (and can be as large as $\Omega(N)$ in general).

Robustness of $\hat{G}_\mathcal{U}$

Svensson argues that if it is possible to satisfy a $1 - \alpha$ fraction of the constraints in $\mathcal{U}$, then there exists a subset $S \subseteq T$ of at most $|S| \leq e_1$ test-vertices such that $\mathsf{depth}_B(\hat{G}_\mathcal{U} - S) \leq d_1$. Similarly, if it is not possible to satisfy a β-fraction of the constraints, then for any subset $S \subseteq T$ of at most $|S| \leq e_2$ test-vertices, we have $\mathsf{depth}_B(\hat{G}_\mathcal{U} - S) \geq d_2$. This does not directly show that $\hat{G}_\mathcal{U}$ is depth-robust since we are not allowed to delete bit-vertices. However, one can easily transform $\hat{G}_\mathcal{U}$ into a graph $G_\mathcal{U}$ on the $N = |T|$ test nodes such that $G_\mathcal{U}$ is (e, d)-depth robust if and only if for all subsets $S \subseteq T$ of $|S| \leq e$ test vertices in $\hat{G}_\mathcal{U}$, we have $\mathsf{depth}_B(\hat{G}_\mathcal{U} - S) \geq d$. It is worth mentioning that we can view these guarantees as a form of *weighted* depth-robustness where all test-vertices have weight 1 and all bit-vertices have weight ∞, i.e., if $1 - \alpha$ fraction of the constraints in $\mathcal{U}$, then we can find a subset S of nodes with weight $\mathsf{weight}(S) \leq e_1$ such that $\mathsf{depth}(\hat{G}_\mathcal{U} - S) \leq d_1$, and if it is not possible to satisfy β-fraction of the constraints, then for any subset S with $\mathsf{weight}(S) \leq e_2$ we have $\mathsf{depth}(\hat{G}_\mathcal{U} - S) \geq d_1$.

Graph Coloring and Robustness

An equivalent way to view the problem of *weighted* reducibility (resp. depth-robustness) is in terms of graph coloring. This view is central to Svensson's argument. In particular, if we can find a depth reducing set $S \subseteq T$ of size $|S| \leq e$ such that $\mathsf{depth}_B(\hat{G}_\mathcal{U} - S) \leq d$, then we can define a d-coloring $\chi : B \to [d]$ of each of the bit-vertices such that the coloring χ is consistent with every remaining test node $v \in T \setminus S$. Here, consistency means that $\max_{b \in \mathsf{parents}(v)} \chi(b) < \min_{b \in \mathsf{children}(v)} \chi(b)$. In fact, it is not too difficult to see that there is a subset $S \subseteq T$ of $|S| \leq e$ test-vertices such that $\mathsf{depth}_B(\hat{G}_\mathcal{U} - S) \leq d$ if and only if there

is a d-coloring χ such that $\left|\{v \ : \ \max_{b \in \mathsf{parents}(v)} \chi(b) \geq \min_{b \in \mathsf{children}(v)} \chi(b)\}\right| \leq e$, i.e., given a d-coloring χ of the bit vertices, we can simply select $S = \{v \ : \ \max_{b \in \mathsf{parents}(v)} \chi(b) \geq \min_{b \in \mathsf{children}(v)} \chi(b)\}$ of inconsistent test-vertices and then for every $u \in B$ we can inductively show that $\mathsf{depth}_B(u, \hat{G}_{\mathcal{U}} - S) \leq \chi(u)$.

Brief Overview of Svensson's Proof

Svensson defines $\chi(w, i)$ to denote the largest color that is smaller than the colors of at least $(1 - \delta)$ fraction of the bit-vertices in $B_{i,w}$, i.e., $\chi(w, i) = \max\{\text{color } c \ : \ \Pr_{b \in B_{i,w}}[\chi(b) \geq c] \geq 1 - \delta\}$. Suppose that it is not possible to satisfy a $\beta = \frac{\delta \eta^2}{t^2 k^2}$-fraction of the constraints in $\mathcal{U}$ for tunable parameters $t, \eta > 0$ that are part of Svensson's construction. The core piece of Svensson's proof is demonstrating that if the set $S = \{v \ : \ \max_{b \in \mathsf{parents}(v)} \chi(b) \geq \min_{b \in \mathsf{children}(v)} \chi(b)\}$ of inconsistent test-vertices has size $|S| \leq (1 - 32\delta)|T|$, then we can find some $w \in W$ such that $\Pr[\chi(w, i) > \chi(w, i + 1)] \geq 32\delta^2$ for some constant c that depends on various parameters of the construction. Svensson notes that by symmetry of the construction $\hat{G}_{\mathcal{U}}$, we can assume without loss of generality that $\chi(i, w) \leq \chi(i + 1, w)$ for any $i \leq L$. We remark that this will *not* necessarily be the case after our indegree reduction step. Thus, it immediately follows that χ uses more than $32|T|\delta^2$ colors, i.e., $\mathsf{depth}_B(u, \hat{G}_{\mathcal{U}} - S) \geq 32|T|\delta^2$.

4.2 Reducing the Indegree

As previously discussed, Svensson's construction has indegree that is too large for the purposes of bounding the pebbling complexity by finding a gap between known results implied by (e_1, d_1)-reducibility and (e_2, d_2)-depth robustness. To perform indegree reduction, we use a γ-extreme depth-robust graph $G_{\gamma, L+1}$ with $L + 1$ vertices in a procedure $\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}})$ to decide which edges in $\hat{G}_{\mathcal{U}}$ to keep and which edges to discard. Intuitively, we will keep the edge $(b_\ell, t_{\ell'})$ from a bit vertex $b_\ell \in B_\ell$ on layer $\ell \leq \ell'$ to test vertex $t_{\ell'} \in T_{\ell'}$ on layer ℓ' if and only if $\ell = \ell'$ or $G_{\gamma, L+1}$ contains the edge (ℓ, ℓ'). Similarly, we will keep the edge $(t_\ell, b_{\ell'})$ from a test vertex $t_\ell \in T_\ell$ on layer $\ell < \ell'$ to bit vertex $b_{\ell'} \in B_{\ell'}$ on layer ℓ' if and only if $(\ell, \ell') \in G_{\gamma, L+1}$. The result is a new DAG $\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}})$ with substantially smaller indegree and outdegree $\mathcal{O}\left(N^\varepsilon \log^2 N\right)$ instead of $\mathcal{O}(N)$.

Transformation $\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}})$

Input: An instance $\hat{G}_{\mathcal{U}} = (V, E)$ of the Svensson's construction, whose vertices are partitioned into $L + 1$ bit-layers $B_0, \ldots, B_L$ and L test-layers $T_0, \ldots, T_{L-1}$, a γ-extreme depth robust graph $G_{\gamma, L+1} = (V_\gamma = [L + 1], E_\gamma)$.

1. Let $G' = (V, E)$ be a copy of $\hat{G}_{\mathcal{U}}$.
2. If $e = (b, t)$ is an edge in G, where $b \in B_i$ and $t \in T_j$, delete e from G' if $i \neq j$ and $(i, j) \notin E_\gamma$.
3. If $e = (t, b)$ is an edge in G, where $b \in B_i$ and $t \in T_j$, delete e from G' if $(j, i) \notin E_\gamma$.

Output: G'

We remark that we only delete edges from $\hat{G}_{\mathcal{U}}$. Thus for any subset $S \subseteq T$ of $|S| \leq e_1$ test vertices, we have $\mathsf{depth}_B(\hat{G}_{\mathcal{U}} - S) \geq \mathsf{depth}_B(\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}}) - S)$. Hence, $\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}})$ is certainly not more depth-robust than $\hat{G}_{\mathcal{U}}$. The harder argument is showing that the graph $\mathsf{Sparsify}_{G_{\gamma, L+1}}(\hat{G}_{\mathcal{U}})$ is still depth-robust when our unique games instance $\mathcal{U}$ has no assignment satisfying a β fraction of the edges.

Assuming that the Unique Games instance is unsatisfiable, Lemma 4 implies that as long as $32\delta^2|T|$ test-vertices are consistent with our coloring, we can find some $w \in W$ such that w is *locally consistent* on at least $32\delta^2 L$ layers, i.e., w is locally consistent on layer ℓ if $\forall \ell' > \ell$ we have $\chi(w, \ell') > \chi(w, \ell)$.

The parameters η, t in Lemma 4 are tunable parameters of the reduction.

▶ **Lemma 4.** *Let χ be any coloring of $\mathsf{Sparsify}_{G_\gamma, L+1}(\hat{G}_{\mathcal{U}})$. If the Unique Games instance has no labeling that satisfies a fraction $\frac{\delta\eta^2}{t^2 k^2}$ of the constraints and at least $32\delta^2|T|$ test vertices are consistent with χ, then there exists $w \in W$ with*

$$\Pr_{\ell \in [L]} [\chi(w, \ell') > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma] \geq 32\delta^2.$$

We remark that the proof of Lemma 4 closely follows Svensson's argument with a few modifications. While the modifications are relatively minor, specifying these modifications requires a complete description of Svensson's construction. We refer an interested reader to Appendix B for details and for the formal proof of Lemma 4.

Lemma 5 now shows that $\mathsf{Sparsify}_{G_\gamma, L+1}(\hat{G}_{\mathcal{U}})$ is still depth-robust in case 2. The main challenge is that after we sparsify the graph, we can no longer assume that $\chi(w, \ell') > \chi(w, \ell)$ for all $\ell' > \ell$ without loss of generality, e.g., even if there are many i's for which $\chi(w, i+1) > \chi(w, i)$ we could have a sequence like $\chi(w, 1) = 1, \chi(w, 2) = 2, \chi(w, 3) = 2, \chi(w, 4) = 2, \chi(w, 5) = 1, \chi(w, 6) = 2, \ldots$. We rely on the fact that $G_{\gamma, L+1}$ is extremely depth-robust to show that for any sufficiently large subset $\mathsf{LC} \subseteq [L]$ of layers for which w is *locally consistent*, there must be a subsequence $\mathsf{P}_w \subseteq \mathsf{LC}$ of length $|\mathsf{P}_w| \geq |\mathsf{LC}| - \gamma L$ over which $\chi(w, \cdot)$ is strictly increasing.

▶ **Lemma 5.** *If the Unique Games instance has no labeling that satisfies a fraction $\frac{\delta\eta^2}{t^2 k^2}$ of the constraints and $\gamma \leq 31\delta^2$, then for every set $S \subseteq T$ of at most $|S| \leq (1 - 32\delta)|T|$ test-vertices the graph $G' = \mathsf{Sparsify}_{G_\gamma, L+1}(\hat{G}_{\mathcal{U}})$ has a path of length $\delta^2 L$.*

Proof. Suppose the Unique Games instance has no labeling that satisfies a fraction $\frac{\delta\eta^2}{t^2 k^2}$ of the constraints. Let S contain at most $(1 - 32\delta)|T|$ and define the labeling $\chi(b) = \mathsf{depth}_B(b, G' - S)$. By Lemma 4, there exists $w \in W$ with

$$\Pr_{\ell \in [L]} [\chi(w, \ell') > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma] \geq 32\delta^2.$$

Let $\mathsf{LC} \subseteq [L]$ denote the subset of layers over which w is locally consistent. We remark that each $\ell \in \mathsf{LC}$ corresponds to a node in $G_{\gamma, L+1}$ and that $G_{\gamma, L+1}[\mathsf{LC}]$ contains a path $\mathsf{P}_w = (\ell_1, \ldots, \ell_k)$ of length $k \geq |\mathsf{LC}| - \gamma L \geq (32\delta^2 - \gamma)L$. We also note that $\chi(w, \ell_{i+1}) > \chi(w, \ell_i)$ for each $i < k$. Hence, $\chi(w, \ell_k) \geq k$, which means that $\mathsf{depth}_B(b, G' - S) \geq (32\delta^2 - \gamma)L \geq \delta^2 L$ as long as $\gamma \leq 31\delta^2$. ◀

Theorem 6, our main technical result in this section, states that it is Unique Games hard to distinguish between (e_1, d_1)-reducible and (e_2, d_2)-depth robust graphs even for a DAG G with N vertices and $\mathsf{indeg}(G) = \mathcal{O}\left(N^\varepsilon \log^2 N\right)$.

▶ **Theorem 6.** *For any integer $k \geq 2$ and constant $\varepsilon > 0$, given a DAG G with N vertices and $\mathsf{indeg}(G) = \mathcal{O}\left(N^\varepsilon \log^2 N\right)$, it is Unique Games hard to distinguish between the following cases: (1) (Completeness): G is $\left(\left(\frac{1-\varepsilon}{k}\right)N, k\right)$-reducible, and (2) (Soundness): G is $\left((1-\varepsilon)N, N^{1-\varepsilon}\right)$-depth robust.*

Proof. Recall that we can transform $G' = \mathsf{Sparsify}_{G_\gamma, L+1}(\hat{G}_\mathcal{U})$ into an unweighted graph G over the $N = |T|$ test-vertices. In particular, we add the edge (u, v) to G if and only if there was a path of length 2 from u to v in G'. We remark that the indegree is $\mathsf{indeg}(G) = \mathcal{O}\left(N^\varepsilon \log^2 N\right)$ and that for any $S \subseteq T$, we have $\mathsf{depth}(G - S) \le \mathsf{depth}_B(G' - S) \le \mathsf{depth}(G - S) + 1$. Completeness now follows immediately from Theorem 20 under the observation that we only removed edges from Svensson's construction. Soundness follows immediately from Theorem 20 and Lemma 5. ◀

Obtaining DAGs with Constant Degree

We can now apply a second indegree reduction procedure $\mathsf{IDR}(G, \gamma)$. For a graph $G = (V, E)$, the procedure $\mathsf{IDR}(G, \gamma)$ replaces each node $v \in V$ with a path $P_v = v_1, \ldots, v_{\delta+\gamma}$, where δ is the indegree of G. For each edge $(u, v) \in E$, we add the edge $(u_{\delta+\gamma}, v_j)$ whenever (u, v) is the jth incoming edge of v, according to some fixed ordering. [5] give parameters e_2 and d_2 so that $\mathsf{IDR}(G, \gamma)$ is (e_2, d_2)-depth robust if G is (e, d)-depth robust. For a formal description of $\mathsf{IDR}(G, \gamma)$, see Appendix B. We complete the reduction by giving parameters e_1 and d_1 so that $\mathsf{IDR}(G, \gamma)$ is (e_1, d_1)-reducible if G is (e, d)-reducible.

▶ **Lemma 7.** *There exists a polynomial time procedure $\mathsf{IDR}(G, \gamma)$ that takes as input a DAG G with N vertices and $\mathsf{indeg}(G) = \delta$ and outputs a graph $G' = \mathsf{IDR}(G, \gamma)$ with $(\delta + \gamma)N$ vertices and $\mathsf{indeg}(G') = 2$. Moreover, the following properties hold: (1) If G is (e, d)-reducible, then $\mathsf{IDR}(G, \gamma)$ is $(e, (\delta + \gamma) \cdot d)$-reducible, and (2) If G is (e, d)-depth robust, then $\mathsf{IDR}(G, \gamma)$ is $(e, \gamma \cdot d)$-depth robust.*

▶ **Corollary 8.** *For any integer $k \ge 2$ and constant $\varepsilon > 0$, given a DAG G with N vertices and maximum indegree $\mathsf{indeg}(G) = 2$, it is Unique Games hard to decide whether G is (e_1, d_1)-reducible or (e_2, d_2)-depth robust for (Completeness): $e_1 = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}$ and $d_1 = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$, and (Soundness): $e_2 = (1 - \varepsilon)N^{\frac{1}{1+2\varepsilon}}$ and $d_2 = 0.9N^{\frac{1+\varepsilon}{1+2\varepsilon}}$.*

5   Putting the Pieces Together

We would now like to apply Theorem 18 and Theorem 19. However, the upper bound on $\mathsf{cc}(G)$ that we obtain from Theorem 18 will not be better than $e_1 N = \frac{1}{k}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$, while the lower bound we obtain from Theorem 19 is just $(1 - \varepsilon)N^{\frac{2+\varepsilon}{1+2\varepsilon}}$, so we do not get our desirable gap between the upper and lower bounds. We therefore discard Theorem 18 and Theorem 19 altogether and instead apply a graph transformation with explicit bounds on pebbling complexity.

▶ **Definition 9** (Superconcentrator). *A graph G with $\mathcal{O}(N)$ vertices is called a superconcentrator if there exists N input vertices, denoted $\mathsf{input}(G)$, and N output vertices, denoted $\mathsf{output}(G)$, such that for all $S_1 \subseteq \mathsf{input}(G), S_2 \subseteq \mathsf{output}(G)$ with $|S_1| = |S_2| = k$, there are k vertex disjoint paths from S_1 to S_2.*

Pippenger gives a superconcentrator construction with depth $\mathcal{O}(\log N)$.

▶ **Lemma 10** ([40]). *There exists a superconcentrator G with at most $42N$ vertices, containing N input vertices and N output vertices, such that $\mathsf{indeg}(G) \le 16$ and $\mathsf{depth}(G) \le \log(42N)$.*

Now we define the overlay of a superconcentrator on a graph G (see Figure 1).

▶ **Definition 11** (Superconcentrator Overlay). *Let $G = (V(G), E(G))$ be a fixed DAG with N vertices and $G_S = (V(G_S), E(G_S))$ be a (priori fixed) superconcentrator with N input vertices $\mathsf{input}(G_S) = \{i_1, \cdots, i_N\} \subseteq V(G_S)$ and N output vertices $\mathsf{output}(G_S) = \{o_1, \cdots, o_N\} \subseteq$*

$V(G_S)$. *We call a graph* $G' = (V(G_S), E(G_S) \cup E_I \cup E_O)$ *a superconcentrator overlay where* $E_I = \{(i_u, i_v) : (u, v) \in E(G)\}$ *and* $E_O = \{(o_i, o_{i+1}) : 1 \le i < N\}$ *and denote as* $G' = \mathsf{superconc}(G)$.

We will denote the interior nodes as $\mathsf{interior}(G') = G' \setminus (\mathsf{input}(G') \cup \mathsf{output}(G'))$ where $\mathsf{input}(G') = \mathsf{input}(G_S)$ and $\mathsf{output}(G') = \mathsf{output}(G_S)$. We remark that when using Pippenger's construction of superconcentrators, it is easy to show that $\mathsf{superconc}(G)$ is $\left(e + \frac{N}{d}, 2d + \log(42N)\right)$-reducible whenever G is (e, d)-reducible, which implies that

$$\mathsf{cc}(\mathsf{superconc}(G)) \le \min_{g \ge d} \left(e + \frac{N}{d}\right) 42N + 2g(42N) + \frac{42N}{g}(2d + \log(42N))\,42N.$$

For more details, we refer an interested reader to Lemma 24 and Corollary 25 in Appendix D. However, these results are not quite as strong as we would like. By comparison, we have the following lower bound on the pebbling complexity from [11]:

$$\mathsf{cc}(\mathsf{superconc}(G)) \ge \min\left(\frac{eN}{8}, \frac{dN}{8}\right).$$

In Lemma 12 we obtain a *significantly tighter* upper bound on $\mathsf{cc}(\mathsf{superconc}(G))$ with an improved pebbling strategy described at the end of this section.

▶ **Lemma 12.** *Let* G *be an* (e, d)-reducible graph with N vertices with $\mathsf{indeg}(G) = 2$. Then $\mathsf{cc}(\mathsf{superconc}(G)) \le \min_{g \ge d} \left\{2eN + 4gN + \frac{43dN^2}{g} + \frac{24N^2 \log(42N)}{g} + 42N \log(42N) + N\right\}.$

With the improved attack in Lemma 12, we can tune parameters appropriately to obtain our main result, Theorem 13.

▶ **Theorem 13.** *Given a DAG* G, *it is Unique Games hard to approximate* $\mathsf{cc}(G)$ *within any constant factor.*

Proof. Let $k \ge 2$ be an integer that we shall later fix and similarly, let $\varepsilon > 0$ be a constant that we will later fix. Given a DAG G with N vertices, then it follows by Corollary 8 that it is Unique Games hard to decide whether G is (e_1, d_1)-reducible or (e_2, d_2)-depth robust for $e_1 = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}$, $d_1 = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$ and $e_2 = (1 - \varepsilon)N^{\frac{1}{1+2\varepsilon}}$ and $d_2 = 0.9N^{\frac{1+\varepsilon}{1+2\varepsilon}}$. If G is (e_1, d_1)-reducible, then by Lemma 12, $\mathsf{cc}(\mathsf{superconc}(G)) \le \min_{g \ge d_2}\{2e_1N + 4gN + \frac{43d_1N^2}{g} + \frac{24N^2 \log(42N)}{g} + 42N \log(42N) + N\}$. Observe that $2e_1N = \frac{2}{k}N^{(2+2\varepsilon)/(1+2\varepsilon)}$, whereas for $g = e_1$ and sufficiently large N, $4gN + \frac{43d_1N^2}{g} + \frac{24N^2 \log(42N)}{g} + 42N \log(42N) + N \le \frac{5}{k}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$. Hence for sufficiently large N,

$$\mathsf{cc}(\mathsf{superconc}(G)) \le \frac{7}{k}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}.$$

On the other hand, if G is (e_2, d_2)-depth robust, then by Lemma 23, $\mathsf{cc}(\mathsf{superconc}(G)) \ge \min\left(\frac{e_2N}{8}, \frac{d_2N}{8}\right)$. Specifically,

$$\mathsf{cc}(\mathsf{superconc}(G)) \ge \frac{e_2N}{8} = \frac{1 - \varepsilon}{8}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}.$$

Let $c > 1$ be any constant. Setting $\varepsilon = 0.1$ and $k = \lceil \frac{560}{9}c^2 \rceil$, we get that if G is (e_1, d_1)-reducible, then $\mathsf{cc}(\mathsf{superconc}(G)) \le \frac{9}{80c^2}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$ but if G is (e_2, d_2)-reducible, then $\mathsf{cc}(\mathsf{superconc}(G)) \ge \frac{9}{80}N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$. Hence, it is Unique Games hard to approximate $\mathsf{cc}(G)$ with a factor of c. ◀
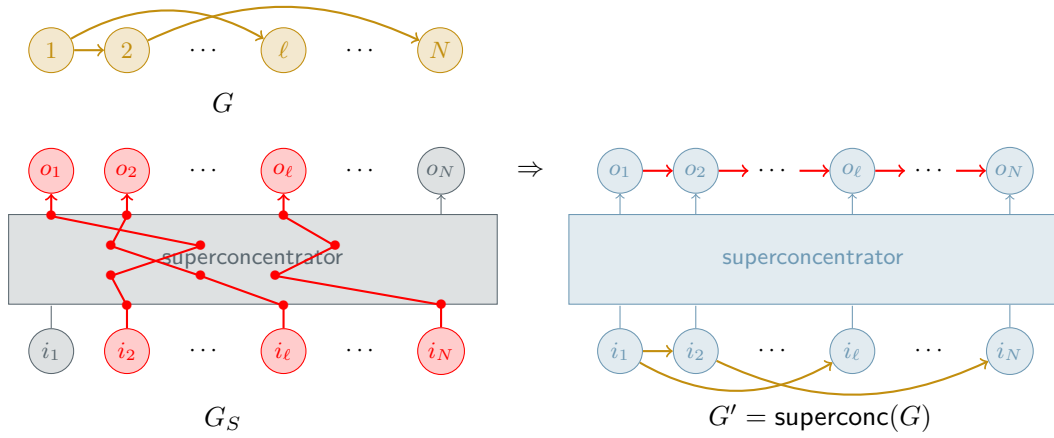
Figure 1 An example of the superconcentrator overlay $G' = \mathsf{superconc}(G)$. We remark that as illustrated in G_S, for any k inputs and k outputs (highlighted in red), there exist k vertex disjoint paths in a superconcentrator.

Improved Pebbling Strategy for $G' = \mathsf{superconc}(G)$

Step 1: Pebble the input nodes $\mathsf{input}(G') = G$.
Step 2: Efficiently pebble $\mathsf{interior}(G')$ using the property of superconcentrator.
Step 3: Pebble all nodes in $\mathsf{output}(G')$ by alternating between light and ballon phases.

- Light Phase - Walk pebble across the interval $I_i = [o_{(i-1)g+1}, o_{ig}]$ in g steps.
  - Precondition for the ith light phase:
  (1) Pebbles on all nodes $v \in \mathsf{parents}(o_{(i-1)g+1})$.
  (2) Pebbles on all nodes $v \in \mathsf{parents}(I_i) \setminus I_i$.
  (3) Pebbles on the set S where S is a (e,d)-depth reducing set for G.
  - Postcondition for the ith light phase:
  (1) Pebbles on the set S and node o_{ig}.
- Balloon Phase - Recover all the missing pebbles in $\mathsf{input}(G') \cup \mathsf{interior}(G')$ for the upcoming light phase.
  - Precondition for the ith balloon phase:
  (1) Pebbles on the set S.
  (2) Pebble on node o_{ig+1} (the first node in the next light phase interval.)
  - Postcondition for the ith balloon phase:
  (1) Pebbles on all nodes in $\mathsf{input}(G') \cup \mathsf{interior}(G') = G' \setminus \mathsf{output}(G')$.
  (2) Pebble on node o_{ig+1} (the first node in the next light phase interval.)

Figure 2 An improved pebbling strategy for $G' = \mathsf{superconc}(G)$. It brought ideas from [2].

Proof of Lemma 12. We will examine the pebbling cost of $\mathsf{superconc}(G)$ for each step shown in Figure 2.

- **Step 1:** We need to place pebbles on all input nodes in G'. By Theorem 18, the pebbling cost of $\mathsf{input}(G') = G$ will be upper bounded by[2]

$$\mathsf{cc}(G) \leq \min_{g \geq d} \left\{ eN + 2gN + \frac{N^2 d}{g} \right\}.$$

- **Step 2:** Start with a configuration with pebbles on every node in $\mathsf{input}(G')$. We have that $\mathsf{depth}(G') \setminus \mathsf{input}(G') = \log(42N)$. Therefore, in time $\log(42N)$, we can place pebbles on every node in $\mathsf{input}(G') \cup \mathsf{interior}(G')$. Hence, the total pebbling cost in Step 2 will be at most $42N \log(42N)$.

- **Step 3:** The goal for step 3 is to walk a pebble across the output nodes starting from o_1 to o_N. To save cost during this step, we should alternate light phases and balloon phases repeatedly N/g times in total since we walk pebble across the interval $I_i = [o_{(i-1)g+1}, o_{ig}]$ of length g in $\mathsf{output}(G')$ in each phase. Let S be a (e, d)-depth reducing set for G. In each light phase, to walk a pebble across the interval I_i, we should keep pebbles on S and $\mathsf{parents}(I_i) \setminus I_i$. Since each node in I_i has two parents outside the interval and we keep one pebble in I_i (the current node) for each step, the maximum number of pebbles to keep would be $|S| + 2g + 1 = e + 2g + 1$ for each step. Hence, the maximum pebbling cost to walk pebble across I_i in i^{th} light phase is $(e + 2g + 1)g$. In each balloon phase, we recover the pebbles in $\mathsf{input}(G') \cup \mathsf{interior}(G')$ for the next light phase. Since S is a (e, d)-depth reducing set, we have that $\mathsf{depth}(G' \setminus (S \cup \mathsf{output}(G'))) \leq d + \log(42N)$. Therefore, recovering the pebbles will cost at most $(d + \log(42N))42N$ for each balloon phase. Hence, the total pebbling cost for Step 3 will be at most $[(e + 2g + 1)g + (d + \log(42N))42N]\frac{N}{g}$.

Taken together, we have that

$$\mathsf{cc}(\mathsf{superconc}_2(G)) \leq \min_{g \geq d} \left\{ \underbrace{eN + 2gN + \frac{N^2 d}{g}}_{\text{Step 1}} + \underbrace{42N \log(42N)}_{\text{Step 2}} + \underbrace{[(e + 2g + 1)g + (d + \log(42N)42N]\frac{N}{g}}_{\text{Step 3}} \right\}$$

$$\leq \min_{g \geq d} \left\{ 2eN + 4gN + \frac{43dN^2}{g} + \frac{24N^2 \log(42N)}{g} + 42N \log(42N) + N \right\}$$

as desired. ◀

---- **References** ----

1 Martín Abadi, Michael Burrows, and Ted Wobber. Moderately Hard, Memory-Bound Functions. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA*, 2003.

2 Joël Alwen and Jeremiah Blocki. Efficiently Computing Data-Independent Memory-Hard Functions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 241–271. Springer, Heidelberg, August 2016. `doi:10.1007/978-3-662-53008-5_9`.

---

[2] Note that Theorem 18 shows the upper bound of the pebbling cost to pebble the last node of G. Here, the difference is that we have to pebble all nodes in $\mathsf{input}(G') = G$, not the last node of G only. However, [2] says that we can recover all nodes concurrently by running one more balloon phase and such cost is already contained in the term $N^2 d/g$. Therefore, we have the same upper bound for $\mathsf{cc}(G)$.

3   Joël Alwen and Jeremiah Blocki. Towards practical attacks on argon2i and balloon hashing. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, pages 142–157. IEEE, 2017.

4   Joël Alwen, Jeremiah Blocki, and Ben Harsha. Practical Graphs for Optimal Side-Channel Resistant Memory-Hard Functions. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1001–1017. ACM Press, October/November 2017. `doi:10.1145/3133956.3134031`.

5   Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-Robust Graphs and Their Cumulative Memory Complexity. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 3–32. Springer, Heidelberg, April/May 2017. `doi:10.1007/978-3-319-56617-7_1`.

6   Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Sustained Space Complexity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 99–130. Springer, Heidelberg, April/May 2018. `doi:10.1007/978-3-319-78375-8_4`.

7   Joël Alwen and Vladimir Serbinenko. High Parallel Complexity Graphs and Memory-Hard Functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 595–603. ACM Press, June 2015. `doi:10.1145/2746539.2746622`.

8   Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential Algorithms for Unique Games and Related Problems. In *51st FOCS*, pages 563–572. IEEE Computer Society Press, October 2010. `doi:10.1109/FOCS.2010.59`.

9   Nikhil Bansal and Subhash Khot. Optimal Long Code Test with One Free Bit. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 453–462, 2009.

10   Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed Locally Correctable Code in Computationally Bounded Channels. In *IEEE International Symposium on Information Theory (ISIT)*, 2019.

11   Jeremiah Blocki, Benjamin Harsha, Siteng Kang, Seunghoon Lee, Lu Xing, and Samson Zhou. Data-Independent Memory Hard Functions: New Attacks and Stronger Constructions. *IACR Cryptology ePrint Archive*, 2018:944, 2018.

12   Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On Locally Decodable Codes in Resource Bounded Channels. *CoRR*, abs/1909.11245, 2019.

13   Jeremiah Blocki, Ling Ren, and Samson Zhou. Bandwidth-Hard Functions: Reductions and Lower Bounds. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1820–1836. ACM Press, October 2018. `doi:10.1145/3243734.3243773`.

14   Jeremiah Blocki and Samson Zhou. On the Depth-Robustness and Cumulative Pebbling Cost of Argon2i. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 445–465. Springer, Heidelberg, November 2017. `doi:10.1007/978-3-319-70500-2_15`.

15   Jeremiah Blocki and Samson Zhou. On the Computational Complexity of Minimal Cumulative Cost Graph Pebbling. *Financial Cryptography and Data Security (FC 2018)*, 2018.

16   Ethan Cecchetti, Ian Miers, and Ari Juels. PIEs: Public incompressible encodings for decentralized storage. Cryptology ePrint Archive, Report 2018/684, 2018. URL: `https://eprint.iacr.org/2018/684`.

17   Ashok K. Chandra. Efficient Compilation of Linear Recursive Programs. In *SWAT (FOCS)*, pages 16–25, 1973.

18   Moses Charikar, Venkatesan Guruswami, and Rajsekar Manokaran. Every Permutation CSP of arity 3 is Approximation Resistant. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 62–73, 2009.

19   Stephen A. Cook. An Observation on Time-storage Trade off. In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, STOC '73, pages 29–33, 1973.

20   Erik D. Demaine and Quanquan C. Liu. Inapproximability of the Standard Pebble Game and Hard to Pebble Graphs. In *Algorithms and Data Structures - 15th International Symposium, WADS 2017, St. John's, NL, Canada, July 31 - August 2, 2017, Proceedings*, pages 313–324, 2017.

21 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 376–389, 2018.

22 Cynthia Dwork, Andrew Goldberg, and Moni Naor. On Memory-Bound Functions for Fighting Spam. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 426–444. Springer, 2003. URL: `http://www.iacr.org/cryptodb/archive/2003/CRYPTO/1266/1266.pdf`.

23 Cynthia Dwork, Moni Naor, and Hoeteck Wee. Pebbling and Proofs of Work. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 37–54. Springer, Heidelberg, August 2005. `doi:10.1007/11535218_3`.

24 Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of Space. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 585–605. Springer, Heidelberg, August 2015. `doi:10.1007/978-3-662-48000-7_29`.

25 Ben Fisch. Tight Proofs of Space and Replication. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 324–348. Springer, Heidelberg, May 2019. `doi:10.1007/978-3-030-17656-3_12`.

26 Christian Forler, Stefan Lucks, and Jakob Wenzel. Memory-Demanding Password Scrambling. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 289–305. Springer, Heidelberg, December 2014. `doi:10.1007/978-3-662-45608-8_16`.

27 John R. Gilbert, Thomas Lengauer, and Robert Endre Tarjan. The Pebbling Problem is Complete in Polynomial Space. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC)*, pages 237–248, 1979.

28 Michel X. Goemans and David P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2SAT. In *26th ACM STOC*, pages 422–431. ACM Press, May 1994. `doi:10.1145/195058.195216`.

29 Carl E. Hewitt and Michael S. Paterson. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, 1970.

30 Jia-Wei Hong and H. T. Kung. I/O complexity: The red-blue pebble game. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA*, pages 326–333, 1981.

31 Subhash Khot. On the power of unique 2-prover 1-round games. In *34th ACM STOC*, pages 767–775. ACM Press, May 2002. `doi:10.1145/509907.510017`.

32 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

33 Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and Grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 576–589, 2017.

34 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

35 Thomas Lengauer and Robert E. Tarjan. Asymptotically Tight Bounds on Time-space Trade-offs in a Pebble Game. *J. ACM*, 29(4):1087–1130, October 1982.

36 Quanquan Liu. Red-Blue and Standard Pebble Games: Complexity and Applications in the Sequential and Parallel Models. Master's thesis, Massachusetts Institute of Technology, February 2017. URL: `http://erikdemaine.org/theses/qliuM.pdf`.

37 Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan. Publicly verifiable proofs of sequential work. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 373–388. ACM, January 2013. `doi:10.1145/2422436.2422479`.

38 Jakob Nordström. Pebble Games, Proof Complexity, and Time-Space Trade-offs. *Logical Methods in Computer Science*, 9(3), 2013.

39 Krzysztof Pietrzak. Proofs of Catalytic Space. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 59:1–59:25. LIPIcs, January 2019. `doi:10.4230/LIPIcs.ITCS.2019.59`.

40 Nicholas Pippenger. Superconcentrators. *SIAM J. Comput.*, 6(2):298–304, 1977.

41 Ling Ren and Srinivas Devadas. Proof of Space from Stacked Expanders. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 262–285, 2016.

42 Ling Ren and Srinivas Devadas. Bandwidth Hard Functions for ASIC Resistance. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 466–492. Springer, Heidelberg, November 2017. `doi:10.1007/978-3-319-70500-2_16`.

43 John E. Savage and Sowmitri Swamy. Space-time trade-offs on the FFT algorithm. *IEEE Transactions on Information Theory*, 24(5):563–568, 1978.

44 John E. Savage and Sowmitri Swamy. Space-Time Tradeoffs for Oblivious Interger Multiplications. In *ICALP*, pages 498–504, 1979.

45 Ola Svensson. Hardness of Vertex Deletion and Project Scheduling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX, and 16th International Workshop, RANDOM. Proceedings*, pages 301–312, 2012.

46 Sowmitri Swamy and John E. Savage. Space-Time Tradeoffs for Linear Recursion. In *POPL*, pages 135–142, 1979.

47 Martin Tompa. Time-space Tradeoffs for Computing Functions, Using Connectivity Properties of Their Circuits. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 196–204, New York, NY, USA, 1978. ACM. `doi:10.1145/800133.804348`.

A   Svensson's Construction

In this section, we review Svensson's Construction. Given an instance $\mathcal{U}$ of Unique Games, Svensson first constructs a weighted instance $\hat{G}_{\mathcal{U}}$ of the DAG reducibility problem. Recall that in the weighted DAG reducibility problem, we are given a DAG G with weights on each node and a target depth d and the goal is to find a minimum weight subset S such that $(G - S)$ contains no path of length d. Given (e_1, d_1) and (e_2, d_2) with $e_1 < e_2$ and $d_1 > d_2$ a weaker goal is simply to distinguish between the following cases: (1) there is a set S of weight at most e_1 s.t. $(G - S)$ contains no path of length d_1, and (2) for all sets S of weight at most e_2 the graph $(G - S)$ contains a path of length d_2. Svensson constructs $\hat{G}_{\mathcal{U}}$ s.t. distinguishing between these cases allows us to solve the original Unique Games instance $\mathcal{U}$.

A.1   Notation

We first review some notation that is used to describe Svensson's initial construction. For $x \in [k]^R$ and a subset S of not necessarily distinct indices of $[R]$, let

$$C_{x,S} = \{z \in [k]^R \,:\, z_j = x_j \forall j \notin S\}$$

denote the sub-cube whose coordinates not in S are fixed according to x. Let

$$C_{x,S,v,w} = \{z \in [k]^R \,:\, z_j = x_{\pi_{v,w}(j)} \forall \pi_{v,w}(j) \notin S\}$$

denote the image of the sub-cube $C_{x,s}$ under $\pi_{v,w}$. Similarly, let

$$C_{x,S}^{\oplus} = \{z \oplus \mathbf{1} \,:\, z \in C_{x,S}\},$$

where $\oplus$ denotes addition modulo k and $\mathbf{1}$ denotes an R-dimensional vector with all elements 1, and let

$$C_{x,S,v,w}^{\oplus} = \{z \oplus \mathbf{1} \,:\, z \in C_{x,S,v,w}\}.$$

A.2   Construction

In Svensson's initial construction, nodes are divided into two sets: bit-vertices and test-vertices. Bit-vertices are assigned weight infinity to guarantee that these nodes are not deleted. Test-vertices are assigned weight one. Here, we focus on the construction of $\hat{G}_{\mathcal{U}}$ though the graph $\hat{G}_{\mathcal{U}}$ is later transformed into an instance $G_{\mathcal{U}}$ of the unweighted DAG reducibility problem, i.e., distinguishing between the cases that $G_{\mathcal{U}}$ is (e_1, d_1)-reducible and (e_2, d_2)-depth robust is sufficient to solve the original Unique Games instance $\mathcal{U}$. See Figure 3 for a simple example of $\hat{G}_{\mathcal{U}}$ along with the transformation to the unweighted instance $G_{\mathcal{U}}$. The DAG $\hat{G}_{\mathcal{U}}$ is defined formally as follows:

- For some L to be fixed, there are $L+1$ layers of bit-vertices. Each bit-layer ℓ with $0 \leq \ell \leq L$ the DAG $\hat{G}_{\mathcal{U}}$ contains bit-vertices $b_{w,x}^{\ell}$ for each $w \in W$ and $x \in [k]^R$. Each bit-vertex is assigned weight ∞.
- There are L layers of test-vertices. For each $0 \leq \ell \leq L-1$, the DAG $\hat{G}_{\mathcal{U}}$ contains test-vertices $t_{x,S,v,w_1,\ldots,w_{2t}}^{\ell}$ for every $x \in [k]^R$, every sequence of indices $S = (s_1, \ldots, s_{\varepsilon R}) \in [R]^{\varepsilon R}$, every $v \in V$ and every sequence $(w_1, \ldots, w_{2t})$ of $2t$ not necessarily distinct neighbors of v. Each test-vertex is assigned weight 1.
- If $\ell \leq \ell'$ and $z \in C_{x,S,v,w_j}$, then there is an edge from bit-vertex $b_{w_j,z}^{\ell}$ to test-vertex $t_{x,S,v,w_1,\ldots,w_{2t}}^{\ell'}$ for each $1 \leq j \leq 2t$.
- If $\ell > \ell'$ and $z \in C_{x,S,v,w_j}^{\oplus}$, then there is an edge from test-vertex $t_{x,S,v,w_1,\ldots,w_{2t}}^{\ell'}$ to bit-vertex $b_{w_j,z}^{\ell}$ for each $1 \leq j \leq 2t$.
- If T is the total number of test-vertices, then L is selected so that $\delta^2 L \geq T^{1-\delta}$.

A.3   Transformation

As mentioned before, in the Svensson's construction, the bit-vertices are given weight ∞ so that they are never deleted, and the graph can be simplified in the following manner without altering the reduction. The transformation to $G_{\mathcal{U}}$ is defined formally as follows:

- For each $0 \leq \ell \leq L-1$, there exists a vertex $v_{x,S,v,w_1,\ldots,w_{2t}}^{\ell}$ for every $x \in [k]^R$, every sequence of indices $S = (s_1, \ldots, s_{\varepsilon R}) \in [R]^{\varepsilon R}$, every $v \in V$ and every sequence $(w_1, \ldots, w_{2t})$ of $2t$ not necessarily distinct neighbors of v.
- If γ is the number of vertices in each layer, then L is selected so that $\delta^2 L \geq (\gamma L)^{1-\delta}$.
- There exists an edge between $v_{x,S,v,w_1,\ldots,w_{2t}}^{\ell}$ and $v_{x',S',v',w_1',\ldots,w_{2t}'}^{\ell'}$ if and only if $\ell < \ell'$ and there exist i, j such that $C_{x,S,v,w_i}^{\oplus} \cap C_{x',S',v',w_j'}$ is nonempty.

▶ **Example 14.** In this example, we will illustrate how to reduce from a Unique Games instance $\mathcal{U}$ to a Svensson's construction $\hat{G}_{\mathcal{U}}$, and a simplification procedure from $\hat{G}_{\mathcal{U}}$ to $G_{\mathcal{U}}$ by examining a simple toy example.

Consider the following Unique Games instance $\mathcal{U} = (G = (V, W, E), [R], \{\pi_{v,w}\}_{v,w})$ with $V = \{v_1\}, W = \{w_1\}, E = \{(v_1, w_1)\}, \pi_{v_1,w_1} : \{1,2\} \to \{2,1\}$, a labeling $\rho : (V \cup W) \to [R]$ such that $\rho(v_1) = 1, \rho(w_1) = 2$, and with the parameters $R = 2, k = 2, t = 1, \delta = 0.1$ and $\epsilon = 0.5$. Then we have the following observations when constructing $\hat{G}_{\mathcal{U}}$:

- Each bit-layer ℓ with $0 \leq \ell \leq L$ contains bit-vertices $b_{w,x}^{\ell}$ for each $w \in W$ and $x \in [k]^R$. Hence, the number of bit-vertices in each layer is $|W| \times |[k]^R| = 1 \times 2^2 = 4$. That is, for each layer i, we have the following bit-vertices:

    $b_{w_1,(11)}^{i}, b_{w_1,(12)}^{i}, b_{w_1,(21)}^{i}$, and $b_{w_1,(22)}^{i}$.

- Each test-layer ℓ with $0 \leq \ell \leq L-1$ contains test-vertices $t^\ell_{x,S,v,w_1,\ldots,w_{2t}}$ for every $x \in [k]^R$, every sequence of indices $S = (s_1,\ldots,s_{\varepsilon R}) \in [R]^{\varepsilon R}$, every $v \in V$ and every sequence $(w_1,\ldots,w_{2t})$ of $2t$ not necessarily distinct neighbors of v. Since $\varepsilon R = 1$ and v_1 has only one neighbor w_1, the number of test-vertices in each layer is $|[k]^R| \times |S| \times |V| \times |N_G(v_1)^{2t}| = 2^2 \times 2 \times 1 \times 1^2 = 8$, where $N_G(v)$ denotes the set of neighbors of v in a graph G. That is, for each layer i, we have the following test-vertices (from now on, we omit the subscript $v, w_1, \ldots, w_{2t}$ in this example since there is only one such case for each test-vertex):

$$t^i_{(11),(1)}, t^i_{(12),(1)}, t^i_{(21),(1)}, t^i_{(22),(1)}, t^i_{(11),(2)}, t^i_{(12),(2)}, t^i_{(21),(2)}, \text{ and } t^i_{(22),(2)}.$$

- There exists an edge from bit vertex $b^\ell_{w_1,z}$ to test-vertex $t^{\ell'}_{x,S}$ if $\ell \leq \ell'$ and $z \in C_{x,S,v_1,w_1}$. We recall that $C_{x,S,v_1,w_1} = \{z \in [k]^R : z_j = x_{\pi_{v_1,w_1}(j)} \; \forall \pi_{v_1,w_1}(j) \notin S\}$. Now it is easy to see that if $S = \{1\}$, $z \in C_{x,S,v_1,w_1}$ if and only if $z_1 = x_2$, and if $S = \{2\}$, $z \in C_{x,S,v_1,w_1}$ if and only if $z_2 = x_1$. Therefore, we have an edge from $b^i_{w_1,(12)}$ to $t^j_{(11),(1)}, t^j_{(21),(1)}, t^j_{(21),(2)}$, and $t^j_{(22),(2)}$ for all $0 \leq i \leq j < L$, and so on.

- There exists an edge from test-vertex $t^{\ell'}_{x,S}$ to bit-vertex $b^\ell_{w_1,z}$ if $\ell > \ell'$ and $z \in C^\oplus_{x,S,v_1,w_1}$, where $\oplus$ denotes addition modulo $k = 2$ and $C^\oplus_{x,S,v_1,w_1} = \{z \oplus 1 : z \in C_{x,S,v_1,w_1}\}$. Hence, for example, if there is an edge from $b^i_{w_1,(12)}$ to $t^j_{x,S}$ then there should be an edge from $t^j_{x,S}$ to $b^{j'}_{w_1,(21)}$ for all $j' > j$ since $(12) \oplus \mathbf{1} = (12) \oplus (11) = (21)$.

When transforming $\hat{G}_\mathcal{U}$ into $G_\mathcal{U}$, we can observe that $C^\oplus_{x,S,v_1,w_1} \cap C_{x',S',v_1,w_1}$ is nonempty if and only if there is a path between two test-vertices through one bit-vertex. Taken together, we have the following structure of graphs reduced from a Unique Games instance $\mathcal{U}$, as shown in Figure 3.   ◄

B   Modified Construction

Given an instance $\hat{G}_\mathcal{U}$ of the Svensson's construction and a γ-extreme depth-robust graph $G_{\gamma,L+1} = (V_\gamma = [L+1], E_\gamma)$, we formally define our modified instance $G' = \mathsf{Sparsify}_{G_{\gamma,L+1}}(\hat{G}_\mathcal{U})$ in the following manner.

Transformation $\mathsf{Sparsify}_{G_{\gamma,L+1}}(\hat{G}_\mathcal{U})$

Input: An instance $\hat{G}_\mathcal{U} = (V, E)$ of the Svensson's construction, whose vertices are partitioned into $L+1$ bit-layers $B_0,\ldots,B_L$ and L test-layers $T_0,\ldots,T_{L-1}$, a γ-extreme depth robust graph $G_{\gamma,L+1} = (V_\gamma = [L+1], E_\gamma)$.

1. Let $G' = (V, E)$ be a copy of $\hat{G}_\mathcal{U}$.
2. If $e = (b, t)$ is an edge in $\hat{G}_\mathcal{U}$, where $b \in B_i$ and $t \in T_j$, delete e from G' if $i \neq j$ and $(i, j) \notin E_\gamma$.
3. If $e = (t, b)$ is an edge in $\hat{G}_\mathcal{U}$, where $b \in B_i$ and $t \in T_j$, delete e from G' if $(j, i) \notin E_\gamma$.

Output: G'

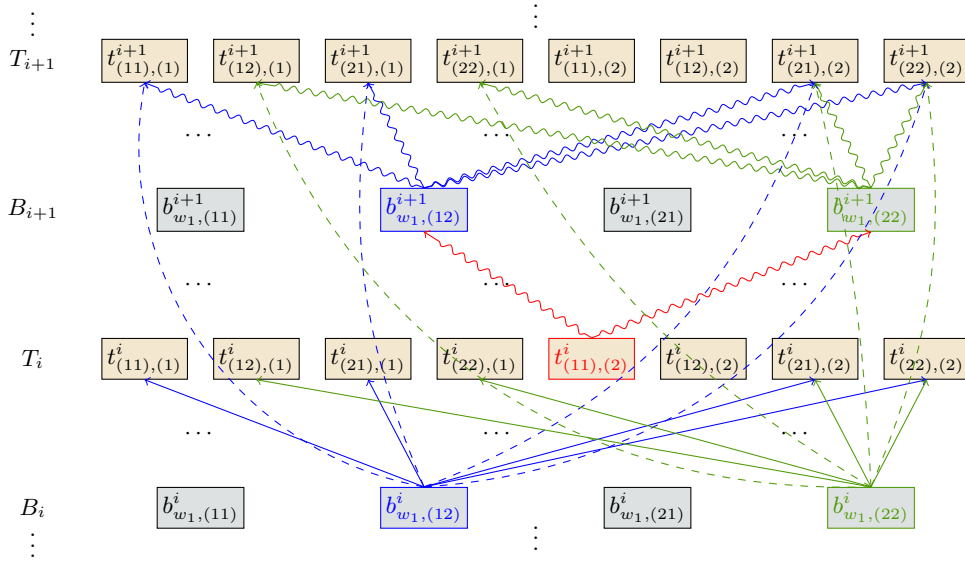We give an illustration of the $\mathsf{Sparsify}$ procedure in Figure 4.

Correspondingly, our modified instance can also be simplified in the following manner without altering the reduction.

- For a input parameter γ, let $G_{\gamma,L+1} = (V_\gamma = [L+1], E_\gamma)$ be an $\frac{\gamma}{2}$-extreme depth robust graph with $L+1$ vertices, which we use $[L+1]$ to represent.
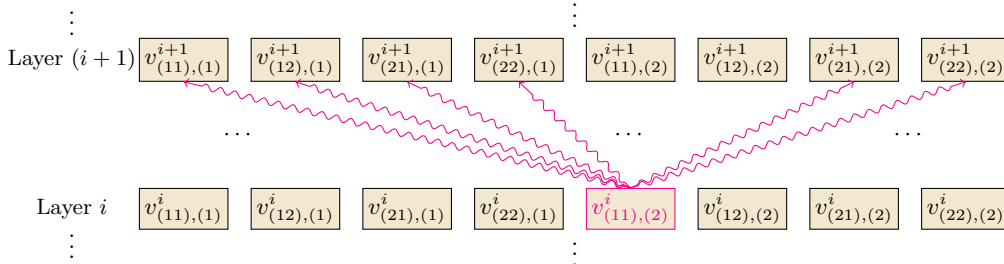
Figure 3 An example of reduction $\hat{G}_{\mathcal{U}}$ from a Unique Games instance $\mathcal{U}$ and a transformation into $G_{\mathcal{U}}$ illustrated in Example 14. We remark that in $\hat{G}_{\mathcal{U}}$ and $G_{\mathcal{U}}$, we only drew edges from the highlighted vertices for simplicity and readability. In $G_{\mathcal{U}}$, we only keep test-vertices from $\hat{G}_{\mathcal{U}}$ and connect two vertices if there is a path between those two through one bit-vertex. Therefore, we can easily check that totally 6 edges (shown in snaked magenta edges) going out from the vertex $v^i_{(11),(2)}$ in $G_{\mathcal{U}}$ comes from the edges $(t^i_{(11),(2)}, b^{i+1}_{w_1,(12)})$ and $(t^i_{(11),(2)}, b^{i+1}_{w_1,(22)})$ (shown in a snaked red edge) and edges from $b^{i+1}_{w_1,(12)}$ and $b^{i+1}_{w_1,(22)}$ to the test layer T_{i+1} (shown in snaked blue/green edges) in $\hat{G}_{\mathcal{U}}$. We also note that the edges starting from the ith bit layer B_i go to every upper test layers T_j for all $j \geq i$. Those edges are represented as dashed ones.
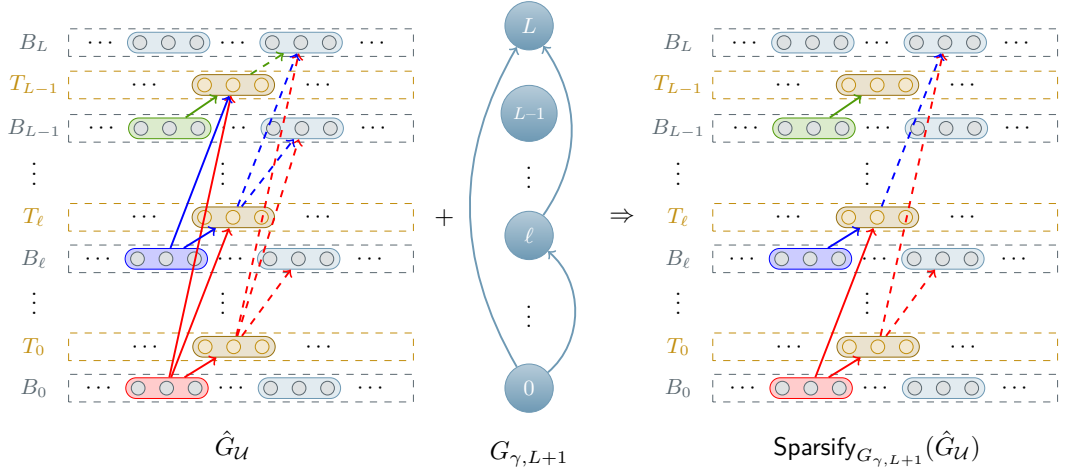
Figure 4 A description of the transformation $\mathsf{Sparsify}_{G_{\gamma,L+1}}(\hat{G}_{\mathcal{U}})$ where $\hat{G}_{\mathcal{U}}$ is Svensson's construction and $G_{\gamma,L+1}$ is a γ-extreme depth-robust graph. We remark that the edges between the subsets of nodes indicate that every node in the input subset is connected to every node in the output subset (in this example, there should be $3 \times 3 = 9$ edges from B_0 to T_0.)

- For each $0 \leq \ell \leq L-1$, there exists a vertex $v^\ell_{x,S,v,w_1,\ldots,w_{2t}}$ for every $x \in [k]^R$, every sequence of indices $S = (s_1, \ldots, s_{\varepsilon R}) \in [R]^{\varepsilon R}$, every $v \in V$ and every sequence $(w_1, \ldots, w_{2t})$ of $2t$ not necessarily distinct neighbors of v.
- If γ is the number of vertices in each layer, then L is selected so that $\delta^2 L \geq (\gamma L)^{1-\delta}$.
- There exists an edge between $v^\ell_{x,S,v,w_1,\ldots,w_{2t}}$ and $v^{\ell'}_{x',S',v',w'_1,\ldots,w'_{2t}}$ if and only if $\ell < \ell'$, the edge (ℓ,ℓ') is in E_γ, and there exist i,j such that $C^\oplus_{x,S,v,w_i} \cap C_{x',S',v',w'_j}$ is nonempty.

We first recall the following definition of influence of the ith coordinate:

$$\mathsf{Infl}_i(f) = \mathbb{E}_x\left[\mathsf{Var}(f)|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_R\right].$$

We now reference the key theorem used in Svensson's analysis.

▶ **Theorem 15** ([31, 45]). *For every $\varepsilon, \delta > 0$ and integer k, there exists $\eta > 0$ and integers t, d such that any collection of functions $f_1, \ldots, f_t : [k]^R \to \{0,1\}$ that satisfies*
- $\forall j, \mathbb{E}[f_j] \geq \delta$
- $\forall i \in [R], \forall 1 \leq \ell_1 \neq \ell_2 \leq t$: $\min\left\{\mathsf{Infl}^d_i(f_{\ell_1}), \mathsf{Infl}^d_i(f_{\ell_2})\right\} \leq \eta$

has

$$\Pr_{x,S_\varepsilon}\left[\bigwedge_{j=1}^t f_j(C_{x,S_\varepsilon}) \equiv 0\right] \leq \delta.$$

We now show that the transformed graph maintains similar properties as Svensson's construction, given an instance of Unique Games. The following statement is analogous to Lemma 4.7 in [45].

▶ **Reminder of Lemma 4.** *Let χ be any coloring of $\mathsf{Sparsify}_{G_{\gamma,L+1}}(\hat{G}_{\mathcal{U}})$. If the Unique Games instance has no labeling that satisfies a fraction $\frac{\delta\eta^2}{t^2 k^2}$ of the constraints and at least $32\delta^2|T|$ test vertices are consistent with χ, then there exists $w \in W$ with*

$$\Pr_{\ell \in [L]}\left[\chi(w,\ell') > \chi(w,\ell) \text{ for all } \ell' > \ell \text{ with } (\ell,\ell') \in E_\gamma\right] \geq 32\delta^2.$$

Proof of Lemma 4. As in [45], an equivalent formulation of the problem is finding a coloring χ in $\{1, 2, \ldots, k\}$ to each bit-vertex to minimize the number of unsatisfied test-vertices. Unlike in [45], we say a test-vertex $t^{\ell}_{x,S,v,w_1,\ldots,w_{2t}}$ is satisfied if

$$\max_{\substack{1 \leq j \leq 2t \\ z \in C_{x,S,v,w_j}}} \chi\left(b^{\ell'}_{w_j,z}\right) < \min_{\substack{1 \leq j \leq 2t \\ z \in C^{\oplus}_{x,S,v,w_j}}} \chi\left(b^{\ell''}_{w_j,z}\right),$$

for all $\ell' \leq \ell < \ell''$ with $(\ell', \ell'') \in E_\gamma$, so that all the predecessors of ℓ are assigned lower colors than the successors of ℓ.

We also define the color $\chi(w, \ell)$ for $w \in W$ and $0 \leq \ell \leq L$ as the maximum color that satisfies

$$\Pr_x\left[\chi(b^{\ell}_{w,x}) \geq \chi(w, \ell)\right] \geq 1 - \delta.$$

For $w \in W$ and $0 \leq \ell \leq L - 1$, define the indicator function $f^{\ell}_w : [k]^R \to \{0, 1\}$ by

$$f^{\ell}_w(x) = \begin{cases} 0 & \text{if } \chi\left(b^{\ell'}_{w,x}\right) > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma, \\ 1 & \text{otherwise.} \end{cases}$$

We call a test-vertex $w \in W$ good in test-layer ℓ if for $\chi(w, \ell') > \chi(w, \ell)$ for every edge of the form (ℓ, ℓ') in the γ-extreme depth-robust graph $G_{\gamma, L+1} = (V_\gamma = [L+1], E_\gamma)$.

▷ **Claim 16.** If the Unique Games instance has no labeling satisfying a fraction $\frac{\delta\eta^2}{t^2k^2}$ of the constraints and a fraction 16δ of the vertices of test-layer ℓ are satisfied, then at least a 2δ fraction of the vertices are good in test-layer ℓ.

Proof. Let A_ℓ be the set of satisfied vertices of test-layer ℓ so that for all $\ell' \leq \ell < \ell''$ with $(\ell', \ell'') \in E_\gamma$, it follows that

$$\Pr_{x,S,v,w_1,\ldots,w_{2t}}\left[\max_{\substack{1 \leq j \leq 2t \\ z \in C_{x,S,v,w_j}}} \chi\left(b^{\ell'}_{w_j,z}\right) < \min_{\substack{1 \leq j \leq 2t \\ z \in C^{\oplus}_{x,S,v,w_j}}} \chi\left(b^{\ell''}_{w_j,z}\right)\right] \geq 16\delta,$$

since at least 16δ fraction of the vertices in A_ℓ are satisfied. We call a tuple $(v, w_1, \ldots, w_{2t})$ good if

$$\Pr_{x,S}\left[\max_{\substack{1 \leq j \leq 2t \\ z \in C_{x,S,v,w_j}}} \chi\left(b^{\ell'}_{w_j,z}\right) < \min_{\substack{1 \leq j \leq 2t \\ z \in C^{\oplus}_{x,S,v,w_j}}} \chi\left(b^{\ell''}_{w_j,z}\right)\right] \geq 8\delta,$$

for all $\ell' \leq \ell < \ell''$ with $(\ell', \ell'') \in E_\gamma$. Observe that at least 8δ fraction of the tuples are good.

From the definition of $\chi(w, \ell')$, we have that $\Pr_x\left[\chi\left(b^{\ell'}_{w,x}\right)\right] \geq 1 - \delta$. Hence for a good tuple, it follows that

$$7\delta \leq \Pr_{x,S}\left[\max_{1 \leq j \leq 2t} \chi(w_j, \ell') < \min_{\substack{1 \leq j \leq 2t \\ z \in C^{\oplus}_{x,S,v,w_j}}} \chi\left(b^{\ell''}_{w_j,z}\right)\right] \leq \Pr_{x,S}\left[\bigwedge_{j=1}^{2t} f^{\ell}_{w_j}(C_{x,S,v,w_j}) \equiv 0\right],$$

for all $\ell' \leq \ell < \ell''$ with $(\ell', \ell'') \in E_\gamma$.

Therefore by Theorem 15, at least one of the following cases holds:

1. more than t of the functions have $\mathbb{E}\left[f_{w_j}^\ell\right] < \delta$ so that $\chi(w_j, \ell') > \chi(w_j, \ell)$ for every edge of the form (ℓ, ℓ') in E_γ, or
2. there exist $1 \le \ell_1 \ne \ell_2 \le t$ and $j \in \mathcal{I}[w_{\ell_1}], j' \in \mathcal{I}[w_{\ell_2}]$ such that $\pi_{v,w_{\ell_1}}(j) = \pi_{v,w_{\ell_2}}(j')$, where

$$\mathcal{I}[w] = \{i \in [R] \ : \ \mathsf{Infl}_i^d(f_w^\ell) \ge \eta\}.$$

If Condition 1 holds for at least $\frac{1}{2}$ of the good tuples, or equivalently a 4δ fraction of all tuples, then at least a 2δ fraction of the test-vertices are good in test-layer ℓ because we can pick a vertex $w_j \in W$ uniformly at random by picking a tuple $(v, w_1, \ldots, w_{2t})$ and then taking one of the vertices $w_1, \ldots, w_{2t}$ at random. Conditioned on the (at least) 2δ probability that the tuple is good and satisfies Condition 1, the probability that $\chi(w_j, \ell') > \chi(w_j, \ell)$ for every edge of the form (ℓ, ℓ') in E_γ for the sampled vertex w_j is at least $\frac{1}{2}$. Therefore, $\Pr_w\left[\chi(w, \ell') > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma\right] \ge 2\delta$, so that at least 2δ fraction of the test-vertices are good in test-layer ℓ.

By way of contradiction, we can show that if Condition 2 were to hold for more than half of the good tuples, the assumption that the Unique Games instance has no labeling satisfying a fraction $\frac{\delta\eta^2}{t^2k^2}$ of the constraints is violated. The argument holds exactly as Claim 4.8 in [45], but we repeat it here for completeness.

For every $w \in W$, let $\rho(w)$ be a random label from $\mathcal{I}[w]$. For every $v \in V$, let w be a random neighbor of v and let $\rho(v) = \pi_{v,w}(\rho(w))$. If Condition 2 holds for half of the good tuples, then a random tuple has this property with at least probability 4δ. Thus with probability at least $\frac{1}{4t^2}$, $w = w_{\ell_1}$ and $w' = w_{\ell_2}$ for w, w' randomly picked from the set $\{w_1, \ldots, w_{2t}\}$. Moreover, [45, 9] observes that with probability at least $\frac{\eta^2}{k^2}$, the labeling procedure defines $j = \rho(w)$ and $j' = \rho(w')$. Hence if Condition 2 holds for half of the good tuples,

$$\Pr_{v,w,w'}\left[\pi_{v,w}(\rho(w)) = \pi_{v,w'}(\rho(w'))\right] \ge \frac{4\delta\eta^2}{4t^2k^2},$$

so that over the randomness of the labeling procedure,

$$\Pr_{(v,w)}\left[\rho(v) = \pi_{v,w}(\rho(w))\right] \ge \frac{\delta\eta^2}{t^2k^2},$$

which contradicts the assumption that the Unique Games instance has no labeling satisfying a fraction $\frac{\delta\eta^2}{t^2k^2}$ of the constraints is violated. $\triangleleft$

Consider a subgraph induced by all bit-vertices and a fraction 32δ of the test-vertices and consider the minimum number of colors required for a coloring χ to satisfy the 32δ fraction of the test-vertices. Since at least 16δ fraction of the test-vertices are good in at least 16δ fraction of the test-layers, then by Claim 16,

$$\Pr_{\ell \in [L], w \in W}\left[\chi(w, \ell') > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma\right] \ge 16\delta \cdot 2\delta = 32\delta^2.$$

Therefore, there exists $w \in W$ with $\Pr_{\ell \in [L]}\left[\chi(w, \ell') > \chi(w, \ell) \text{ for all } \ell' > \ell \text{ with } (\ell, \ell') \in E_\gamma\right] \ge 32\delta^2$.   ◀

Interestingly, we can exactly compute the pebbling complexity of the simplified Svensson's construction, when the graph is only represented with the test-vertices.

▶ **Lemma 17.** *Given a (simplified) Svensson's construction $G_{\mathcal{U}}$ that consists of N vertices partitioned across L layers, $\mathsf{cc}(G_{\mathcal{U}}) = \frac{N(L+1)}{2}$.*

Proof. We first show that the pebbling complexity of $G_{\mathcal{U}}$ is at least $\frac{N(L+1)}{2}$. Observe that the Svensson's construction contains $\frac{N}{L}$ vertices in each layer and furthermore, for each pair of layers i and j, there is a perfect matching between vertices of layer i and vertices of layer j among the edges connecting layers i and j. Let $\mathcal{M}_{i,j}$ be the subset of edges between i and j that is perfect matching.

For a given pebble u in layer j, let v_i be the vertex in layer i matched to u by $\mathcal{M}_{i,j}$. To pebble a vertex u in layer j, all of its parents must contain pebbles in the previous round. Namely, there must be a pebble on v_i for all $1 \leq i < j$ in the previous round. Since each $\mathcal{M}_{i,j}$ is a perfect matching, there must be $j-1$ pebbles on the graph solely for the purpose of pebbling node u in layer j. Thus pebbling each node in layer j induces a pebbling cost of at least $j-1$. Since there are $\frac{N}{L}$ pebbles in each layer and L layers, then the total pebbling cost is at least $\frac{N}{L}\sum_{j=1}^{L}(j-1) = \frac{N(L+1)}{2}$, which lower bounds $\mathsf{cc}(G_{\mathcal{U}})$.

On the other hand, consider the natural pebbling where all the pebbles in layer j are pebbled in round j, and no pebble is ever removed. Then the graph is completely pebbled in L rounds, since layer L is pebbled in round L. Moreover, the cost of pebbling round j is $\frac{N}{L}(j-1)$. Hence, the pebbling cost is $\frac{N}{L}\sum_{j=1}^{L}(j-1) = \frac{N(L+1)}{2}$, which upper bounds $\mathsf{cc}(G_{\mathcal{U}})$. ◀

Finally, we give a formal description of the procedure $\mathsf{IDR}(G, \gamma)$. Recall that $\mathsf{IDR}(G, \gamma)$ for a graph $G = (V, E)$ replaces each vertex $v \in V$ with a path $P_v = v_1, \ldots, v_{\delta+\gamma}$, where δ is the indegree of G. For each edge $(u, v) \in E$, we add the edge $(u_{\delta+\gamma}, v_i)$ whenever (u, v) is the ith incoming edge of v, according to some fixed ordering. [5] give parameters e_2 and d_2 so that $\mathsf{IDR}(G, \gamma)$ is (e_2, d_2)-depth robust if G is (e, d)-depth robust. We complete the reduction by giving parameters e_1 and d_1 so that $\mathsf{IDR}(G, \gamma)$ is (e_1, d_1)-reducible if G is (e, d)-reducible.
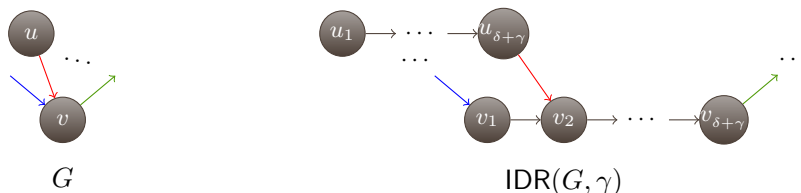

Transformation $\mathsf{IDR}(G, \gamma)$

Input: An DAG $G = (V, E)$ with indegree δ, parameter γ.

1. Let the vertices of G be $[|V|]$.
2. Initialize G' to be a graph with $(\delta+\gamma)|V|$ vertices and let these vertices be $[(\delta+\gamma)|V|]$
3. If $(\delta+\gamma)n + 1 \leq u < (\delta+\gamma)n$ for some integer n, add edge $(u, u+1)$ to G'.
4. If (u, v) is the ith incoming edge of v by some fixed predetermined ordering, then add $(u_{\delta+\gamma}, v_i)$ to G'.

Output: G'

We given an illustration of the IDR transformation in Figure 5.

Figure 5 An example of the transformation $\mathsf{IDR}(G, \gamma)$. We remark that if the red edge (u, v) is the 2nd incoming edge of v, then in $\mathsf{IDR}(G, \gamma)$ we should add $(u_{\delta+\gamma}, v_2)$ to G'.

C   Useful Theorems

We rely on the following results in our constructions and proofs.

▶ **Theorem 18** ([2]). *Let G be a DAG with N vertices and indegree δ. If G is (e, d)-reducible, then $\mathsf{cc}(G) \leq \min_{g \geq d} \left\{ eN + \delta g N + \frac{N^2 d}{g} \right\}$.*

▶ **Theorem 19** ([5]). *Let G be a DAG with N vertices and indegree δ. If G is (e, d)-depth robust, then $\mathsf{cc}(G) \geq ed$.*

While Theorem 18 and Theorem 19 are nice results that relate the pebbling complexities of (e_1, d_1)-reducible and (e_2, d_2)-depth robust graphs, these statements are ultimately misleading in that $d \leq N$ and thus there will never be a gap between the pebbling complexities of the graphs.

▶ **Theorem 20** ([45]). *For any integer $k \geq 2$ and constant $\varepsilon > 0$, given a DAG G with N vertices, it is Unique Games hard to distinguish between the following cases:*

- *(Completeness): G is $\left(\left(\frac{1-\varepsilon}{k} \right) N, k \right)$-reducible.*
- *(Soundness): G is $\left((1 - \varepsilon)N, N^{1-\varepsilon} \right)$-depth robust.*

▶ **Definition 21.** *Given a parameter $0 < \gamma < 1$, a DAG $G = (V, E)$ is γ-extreme depth-robust if G is (e, d)-depth robust for any e, d such that $e + d \leq (1 - \gamma)N$.*

▶ **Theorem 22** ([6]). *For any fixed $0 < \gamma < 1$, there exists a constant $c_1 > 0$ such that for all integers $N > 0$, there exists an γ-extreme depth robust graph G with N vertices and $\mathsf{indeg}(G), \mathsf{outdeg}(G) \leq c_1 \log N$.*

▶ **Lemma 23** ([11]). *Let G be an (e, d)-depth robust graph with N vertices. Then*

$$\mathsf{cc}(\mathsf{superconc}(G)) \geq \min \left(\frac{eN}{8}, \frac{dN}{8} \right).$$

D   Missing Proofs

▶ **Reminder of Lemma 7.** *There exists a polynomial time procedure $\mathsf{IDR}(G, \gamma)$ that takes as input a DAG G with N vertices and $\mathsf{indeg}(G) = \delta$ and outputs a graph $G' = \mathsf{IDR}(G, \gamma)$ with $(\delta + \gamma)N$ vertices and $\mathsf{indeg}(G') = 2$. Moreover, the following properties hold: (1) If G is (e, d)-reducible, then $\mathsf{IDR}(G, \gamma)$ is $(e, (\delta + \gamma) \cdot d)$-reducible, and (2) If G is (e, d)-depth robust, then $\mathsf{IDR}(G, \gamma)$ is $(e, \gamma \cdot d)$-depth robust.*

Proof of Lemma 7. Alwen et al. [5] show that $\mathsf{IDR}(G, \gamma)$ is $(e, \gamma \cdot d)$-depth robust if G is (e, d)-depth robust. It remains to show that $\mathsf{IDR}(G, \gamma)$ is $(e, (\delta + \gamma) \cdot d)$-reducible if G is (e, d)-reducible.

Given an (e, d)-reducible graph $G = (V, E)$ of N vertices, we use $[N]$ to represent the vertices of G and let $G' = \mathsf{IDR}(G, \gamma)$ so that the vertices of G' can be associated with $[(\delta + \gamma)N]$. Let S be a set of e vertices in G such that $\mathsf{depth}(V - S) < d$. Let S' be a set of e vertices in G' so that $(\delta + \gamma)v \in S'$ for each vertex $v \in S$.

Suppose, by way of contradiction, that there exists a path P' of length $(\delta + \gamma) \cdot d$ in $G' - S'$. Observe that if $y, z \in G'$ such that $(\delta + \gamma)a + 1 \leq y \leq (\delta + \gamma)(a + 1)$ and $(\delta + \gamma)b + 1 \leq z \leq (\delta + \gamma)(b + 1)$ for integers $a < b$, then y cannot be connected to z unless $y = (\delta + \gamma)(a + 1)$. Hence that if P' contains vertex $u \in G'$ such that $(\delta + \gamma)c + 1 \leq u \leq (\delta + \gamma)(c + 1)$ and u

is not one of the final $\delta + \gamma - 1$ vertices of P', then $(\delta + \gamma)(c + 1) \in P'$. Thus by a simple Pigeonhole argument, there exists at least d integers $j_1, j_2, \ldots, j_d$ such that $(\delta + \gamma)j_n \in P'$ for $1 \le n \le d$ and moreover there exists an edge in P' from each vertex $(\delta + \gamma)j_n$ to some vertex w such that $(\delta + \gamma)(j_{n+1} - 1) + 1 \le w \le (\delta + \gamma)j_{n+1}$ for $1 \le n \le d - 1$. However, this implies that $j_1, \ldots, j_d$ is a path in G by construction of $\mathsf{IDR}(G, \gamma)$. Moreover, since $(\delta + \gamma)v \in S'$ for each vertex $v \in S$, this implies that $j_1, \ldots, j_d$ is a path of length d in $G - S$, which contradicts the assumption that $\mathsf{depth}(G - S) < d$. ◄

▶ **Reminder of Corollary 8.** *For any integer $k \ge 2$ and constant $\varepsilon > 0$, given a DAG G with N vertices and maximum indegree $\mathsf{indeg}(G) = 2$, it is Unique Games hard to decide whether G is (e_1, d_1)-reducible or (e_2, d_2)-depth robust for (Completeness): $e_1 = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}$ and $d_1 = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$, and (Soundness): $e_2 = (1 - \varepsilon)N^{\frac{1}{1+2\varepsilon}}$ and $d_2 = 0.9N^{\frac{1+\varepsilon}{1+2\varepsilon}}$.*

Proof of Corollary 8. Suppose G' is a graph with M vertices. By applying Lemma 7 to Theorem 6 and setting $k = M^{2\varepsilon}$ and $\gamma = M^{2\varepsilon} - \delta$, then we start from a graph with M vertices and end with a graph G with $N = M^{1+2\varepsilon}$ vertices or equivalently, $M = N^{1/(1+2\varepsilon)}$. Thus, $G = \mathsf{IDR}(G', \gamma)$ is (e, d)-reducible for $e = \frac{(1-\varepsilon)M}{k} = \frac{1-\varepsilon}{k}N^{1/(1+2\varepsilon)}$ and $d = kM^{2\varepsilon} = kN^{2\varepsilon/(1+2\varepsilon)}$. Since $e < \frac{M}{k}$, it is clearly the case that $G = \mathsf{IDR}(G', \gamma)$ is (e_1, d_1)-reducible for $e_1 = \frac{M}{k} > e$ and $d_1 = d = kN^{2\varepsilon/(1+2\varepsilon)}$ as we delete more nodes and the depth reducibility guarantees the same upper bound of the remaining depth. On the other hand, $\mathsf{IDR}(G', \gamma)$ is (e_2, d_2)-depth robust for $e_2 = (1 - \varepsilon)M = (1 - \varepsilon)N^{1/(1+2\varepsilon)}$, while $d_2 = \gamma M^{1-\varepsilon} = (M^{2\varepsilon} - \delta)M^{1-\varepsilon}$. By Theorem 6, $\delta = \mathcal{O}\left(M^\varepsilon \log^2 M\right)$ so that for sufficiently large M, $d_2 = 0.9M^{1+\varepsilon} = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$. ◄

▶ **Lemma 24.** *If G is (e, d)-reducible, then $\mathsf{superconc}(G)$ is $\left(e + \frac{N}{d}, 2d + \log(42N)\right)$-reducible, where N is the number of vertices in $\mathsf{superconc}(G)$.*

Proof. Let $G = (V, E)$ be a (e, d)-reducible DAG with N vertices. Let $G' = \mathsf{superconc}(G)$ and suppose G' has M vertices, which we designate $[M]$. Thus, there exists a set $S \subseteq V$ such that $|S| \le e$ and $\mathsf{depth}(G - S) < d$. Let T be the set of $\frac{N}{d}$ vertices $\{M, M - d, M - 2d, \ldots, M - N + d\}$, so that $T \subseteq \mathsf{output}(G')$. We claim $\mathsf{depth}(G' - S - T) < 2d + \log(42N)$.

Suppose by way of contradiction that there exists a path P in $G' - S - T$ of length at least $2d + \log(42N)$. By Lemma 10, the depth of any path from an input node to an output vertex is at most $\log(42N)$. Moreover, all edges added in the superconcentrator overlay are either between input vertices or two output vertices. Hence, then at least $2d$ vertices of P have to lie in either the first N vertices or the last N vertices of G'. Because P does not contain vertices of T, there is no path of length of length d in the last N vertices of G', so there must be a path of length d in the first N vertices of G', which contradicts $\mathsf{depth}(G - S) < d$.

Therefore, G' is $\left(e + \frac{N}{d}, 2d + \log(42N)\right)$-reducible. ◄

From Lemma 24 we immediately obtain an upper bound on the pebbling complexity of $\mathsf{superconc}(G)$ by applying Theorem 18 to Lemma 24. However, the upper bound is not as strong as we would like.

▶ **Corollary 25.** *Let G be an (e, d)-reducible graph with N vertices. Then*

$$\mathsf{cc}(\mathsf{superconc}(G)) \le \min_{g \ge d} \left(e + \frac{N}{d}\right) 42N + 2g(42N) + \frac{42N}{g}(2d + \log(42N)) 42N.$$