Giving Some Slack: Shortcuts and Transitive Closure Compressions

Shimon Kogan ✉
Weizmann Institute of Science, Rehovot, Israel

Merav Parter ✉
Weizmann Institute of Science, Rehovot, Israel

Abstract

We consider the fundamental problems of reachability shortcuts and compression schemes of the transitive closure (TC) of n-vertex directed acyclic graphs (DAGs) G when we are allowed to neglect the distance (or reachability) constraints for an ϵ fraction of the pairs in the transitive closure of G, denoted by $TC(G)$.

Shortcuts with Slack. For a directed graph $G = (V, E)$, a d-reachability shortcut is a set of edges $H \subseteq TC(G)$, whose addition decreases the directed diameter of G to be at most d. We introduce the notion of shortcuts with *slack* which provide the desired distance bound d for all but a small fraction ϵ of the vertex pairs in $TC(G)$. For $\epsilon \in (0, 1)$, a (d, ϵ)-shortcut $H \subseteq TC(G)$ is a subset of edges with the property that $\text{dist}_{G\cup H}(u, v) \leq d$ for at least $(1 - \epsilon)$ fraction of the (u, v) pairs in $TC(G)$. Our constructions hold for any DAG G and their size bounds are parameterized by the *width* of the graph G defined by the smallest number of directed paths in G that cover all vertices in G.

- For every $\epsilon \in (0, 1]$ and integer $d \geq 5$, every n-vertex DAG G of width ω admits a (d, ϵ)-shortcut of size $\widetilde{O}(\omega^2/(\epsilon d)+n)$. A more delicate construction yields a $(3, \epsilon)$-shortcut of size $\widetilde{O}(\omega^2/(\epsilon d)+n/\epsilon)$, hence of linear size for $\omega \leq \sqrt{n}$. We show that without a slack (i.e., for $\epsilon = 0$), graphs with $\omega \leq \sqrt{n}$ cannot be shortcut to diameter below $n^{1/6}$ using a linear number of shortcut edges.
- There exists an n-vertex DAG G for which any $(3, \epsilon = 1/2^{\sqrt{\log \omega}})$-shortcut set has $\Omega(\omega^2/2^{\sqrt{\log \omega}} + n)$ edges. Hence, for $d = \widetilde{O}(1)$, our constructions are almost optimal.

Approximate TC Representations. A key application of our shortcut's constructions is a $(1-\epsilon)$-approximate *all-successors* data structure which given a vertex v, reports a list containing $(1 - \epsilon)$ fraction of the successors of v in the graph. We present a $\widetilde{O}(\omega^2/\epsilon + n)$-space data structure with a near linear (in the output size) query time. Using connections to Error Correcting Codes, we also present a near-matching space lower bound of $\Omega(\omega^2 + n)$ bits (regardless of the query time) for constant ϵ. This improves upon the state-of-the-art space bounds of $O(\omega \cdot n)$ for $\epsilon = 0$ by the prior work of Jagadish [ACM Trans. Database Syst., 1990].

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Reachability Shortcuts, Width, DAG

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.79

Funding This project is funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 949083), and by the Israeli Science Foundation (ISF), grant No. 2084/18.

1 Introduction

Finding succinct directed graph representations is fundamental graph problem that admits a large collection of applications to databases systems, evolutionary computation, program testing, communication networks and parallel computation [3, 14, 27, 25, 15]. These problems can be subdivided into two (orthogonal) types: *augmentation* and *reduction*. In augmentation

32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 79; pp. 79:1–79:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

problems, the goal is to add a small number of edges to the graph to improve some of the graph's properties. Classical examples are *reachability shortcuts* and *hopsets* whose addition to the graph reduces the number of hops along directed shortest paths. Reduction problems aim at reducing the space of the graph representation, e.g., by computing a sparse subgraph or a low-space data-structure, that preserve some key desired properties of the original graph. Notable examples are transitive closure (TC) reduction and reachability preservers.

In this work, we extend the normal definitions of digraph representation problems with a slack parameter ϵ, which allows us to ignore up to an ϵ-fraction of the TC pairs, and provide the desired reachability (or distance) guarantee for the remaining pairs. We address this slack notion for the settings of reachability shortcuts (an augmentation problem) and TC-reduction and compression schemes (which are reduction problems). Our algorithms hold for the family of n-vertex directed acyclic graphs (DAGs) and are in-particular efficient for graphs of bounded width. The *width* of a DAG G is the minimum number of chains[1] that cover all the vertices of G. This graph parameter has been shown to play an important role in the space bound of reachability shortcuts [18] and TC-compression [14], in the standard setting, where no slack is allowed. A natural question to ask is whether adding a slack parameter can significantly reduce the space bounds of these structures. In this paper we address this question from upper and lower bounds perspectives.

There has been a series of papers studying the concept of slack, mainly in the context of metric embedding [16] and distance compression in undirected graphs [1, 2]. Chan, Dinitz and Gupta [6] studied the notion of spanners with slack, i.e., where the small stretch guarantee[2] holds for $(1-\epsilon)$ fraction of the vertex pairs. Distance oracles and routing schemes with slack have been studied by [2] and [10], respectively. To the best of our knowledge, no prior work addressed the setting of slack in the context of succinct *directed* graph representations.

Reachability Shortcuts. A d-reachability shortcut of a digraph $G = (V, E)$ with transitive closure $TC(G)$ is a set of edges $H \subseteq TC(G)$ such that the (u, v)-shortest path distance in $G \cup H$ is at most d, for every $(u, v) \in TC(G)$. Reachability shortcuts have attracted a lot of attention in the recent years. The key question addressed is what's the best possible distance bound d achievable with a linear number (i.e., $O(|V|)$) of shortcut edges. Recent work demonstrated an upper bound of $d = \widetilde{O}(n^{1/3})$ by Kogan and Parter [17] and a lower bound of $\widetilde{\Omega}(n^{1/4})$ by Bodwin and Hoppenworth [4]. Narrowing this gap is one of the most intriguing open problems in this setting. As we explain in Sec. 2, a key barrier for improving upon the Kogan-Parter [17] result is the fact that the reachability relations between pairs of paths P_1, P_2 *cannot* be captured by adding a small number of shortcut edges between $V(P_1)$ and $V(P_2)$. We therefore start by asking if this barrier can be bypassed when we are allowed to neglect some fraction of the pairs in $V(P_1) \times V(P_2)$, and answer this question in the affirmative.

We introduce the notion of *shortcuts with slack* in which the distance guarantee fails to hold for a small fraction of the pairs in $TC(G)$. Formally, a (d, ϵ) shortcut H satisfies that $\text{dist}_{G\cup H}(u, v) \leq d$ for $(1-\epsilon)$ fraction of the (u, v) pairs in $TC(G)$. We also consider a stronger slack notion where the desired distance guarantee $\text{dist}_{G\cup H}(u, v) \leq d$ holds for $(1-\epsilon)$ fraction of all vertices that are reachable from u, for every[3] u. This notion finds applications in the context of single-source reachability in the parallel and distributed settings [20]. By

---

[1] A chain is a dipath in the transitive closure $TC(G)$.

[2] The stretch of a (standard) spanner $H \subseteq G$ is t if $\text{dist}_H(u, v) \leq t \cdot \text{dist}_G(u, v)$ for every $u, v \in V$.

[3] Symmetrically, one can provide this property for $(1-\epsilon)$ fraction of all the incoming $TC(G)$-neighbors of v, for every v. This can be done by working on a graph in which all the directions of G-edges are reversed.

adding a (d,ϵ) shortcut H to the graph G, one can detect $(1-\epsilon)$ fraction of all reachable vertices from the given source in parallel depth of d and work that depends linearly in $|G \cup H|$.

Shortcuts with slack can also be used to compute shortcuts with a small *average* distance. For a given directed graph G', denote its average distance by:

$$\text{AVdist}(G') = \frac{\sum_{(u,v)\in TC(G')} \text{dist}_{G'}(u,v)}{|TC(G')|}.$$

A set $H \subseteq TC(G)$ is a *d-average distance* shortcut for G if $\text{AVdist}(G \cup H) \leq d$. This notion has been studied in the context of spanners (for undirected graphs) by Chan, Dinitz and Gupta [6] and low-stretch spanning trees [22].

Shortcut Lower Bounds. Standard shortcuts (with no slack) have been studied from a lower bound perspective, as well [25, 12, 13, 21, 4, 19, 28]. For example, Huang and Pettie [13] exhibited a construction of an n-vertex graph for which any $O(n)$-size shortcut set cannot reduce the diameter to below $\Theta(n^{1/6})$. This lower bound has been recently improved by Bodwin and Hoppenworth [4] to $\Omega(n^{1/4})$, and further generalized and simplified by Vassilevska Williams, Xu and Xu [28]. It is noteworthy that the width of these lower bound graphs might be $\Omega(n^{3/4})$. Since bounded width graphs enjoy an improved diameter bounds (as shown in [18]), it is intriguing to characterize the diameter and size tradeoff as a function of the width of the graph. We note that the current lower bound constructions do not hold in the slack setting, hence lower bounds for shortcuts with slack call for new ideas.

Compression Schemes of the Transitive Closure. The succinct storage of the transitive closure and reachability information is a fundamental graph problem that has many applications, in particular, in database systems [29, 8, 30, 14]. A common primitive repeatedly employed in these systems is that of determining whether there exists a directed path between a pair of vertices in the given directed graph G, or determining all vertices that are reachable from a particular vertex. This finds applications in expert systems, relational algebra, object oriented and semantic data models, e.g., for finding all inherited properties in objected-oriented [26] and semantic data models [3].

In an influential work on algorithmic databases, Jagadish [14] introduced the *all-successor* data-structure, which upon a query vertex u reports the list of all vertices that are reachable from u in G. For an n-vertex ω-width DAG, the presented data-structure of Jagadish [14] has space of $\widetilde{O}(\omega \cdot n)$ and near-optimal query time. As this space bound can also be shown to be tight, we again ask whether one can improve the space requirements at the expense of introducing an ϵ slack, where it is allowed to report $(1-\epsilon)$ fraction of the successors of u. Cohen [7] addressed the notion of transitive-closure with slack from a computational time, rather then space, perspective. Specifically, [7] presented a linear time randomized algorithm for computing a $(1-\epsilon)$ approximation of the number of vertices reachable from every vertex. The question that we address in this paper is concerned with compression: what's the minimal space required in order to be able to retrieve a $(1-\epsilon)$ approximation of the transitive closure (or even deducing $(1-\epsilon)$ fraction of the successor lists).

▶ Remark. There are two main differences between TC compression and shortcuts. TC compression does not account for the number of hops along shortest paths, and in this sense its space bounds might be smaller than that required by shortcuts. Shortcuts, on the other hand, only account for the number of added shortcut edges, and the size of the graph comes for free. In this sense, shortcuts might be sparser. Interestingly, we show an almost matching space bounds for these two problems when $d = 3$.

1.1 New Results

All our results hold for n-vertex DAG $G = (V, E)$ of width ω and m edges.

1.1.1 Shortcuts with Slack

We provide new constructions for (d, ϵ)-shortcuts, for any d. For the values of $d = 2$ and $d = 3$, we also provide (almost) matching lower bounds. Our starting observation indeed shows that a 2-shortcut with slack requires $\Omega(\omega \cdot n)$ edges, and this is tight.

▶ **Lemma 1.1** ($(2, \epsilon)$-Shortcuts are Dense)**.** *Every n-vertex DAG admits a $(2, \epsilon)$-shortcut of size $\widetilde{O}(\omega n + n/\epsilon)$ edges. .*

We show that bound of Lemma 1.1 is nearly *tight*:

▶ **Theorem 1.2** ($(2, \epsilon)$-shortcuts)**.** *For every sufficiently large n and positive integer $k \in \mathbb{N}_{\leq n}$, there exists an n-vertex DAG of diameter 3 and width $\omega = \Theta(k)$, such that any $(2, \epsilon = 1/2)$-shortcut requires $\Omega(n \cdot \omega)$ edges.*

For 3-shortcuts, we provide improved bounds for every $\omega = o(n)$:

▶ **Theorem 1.3.** *For every n-vertex ω-width DAG $G = (V, E)$ and $\epsilon \in (0, 1)$, one can compute in $\widetilde{O}(\omega \cdot m/\epsilon + m^{1+o(1)})$ time, a $(3, \epsilon)$-shortcut with $\widetilde{O}((\omega^2 + n)/\epsilon)$ edges.*

We complement this by two lower bound results. A size lower bound of $\Omega(n)$ can be easily obtained for a n-vertex dipath. We thus focus on matching the ω^2 term. Our first lower bound holds for $\epsilon = 2^{-\sqrt{\log \omega}}$ and any width value ω. The value of ϵ is limited by the current upper bounds on the Ruzsa-Szemerédi (RS) number [24]. For an integer n, $RS(n)$ is the *largest* value such that any graph that can be partitioned into n induced matching has at most $n^2/RS(n)$ edges. We use the fact that $RS(n) = 2^{O}(\sqrt{\log n})$ to show:

▶ **Theorem 1.4.** *For every sufficiently large n and an integer $k \leq n/5$, there exists an n-vertex DAG G of diameter 4 and width $\omega = \Theta(k)$, such that every $(3, \epsilon)$-shortcut H for G with $\epsilon = \left(\frac{1}{2}\right)^{\Theta\left(\sqrt{\log \omega}\right)}$ requires $\Omega(\omega^2/2^{O\left(\sqrt{\log \omega}\right)})$ edges.*

For $\epsilon = \left(\frac{1}{2}\right)^{\Theta\left(\sqrt{\log \omega}\right)}$ this almost matches the upper bound of Theorem 1.3.

We also propose an alternative construction that provides a near-optimal size bound for any *constant* value of ϵ, provided that the width of the graph is bounded by $\omega = O(n^{2/3})$. The construction is probabilistic, and it is based on embedding a random $G(n, p)$ graph, rather than the Ruzsa-Szemerédi graph (as used in Thm. 1.4).

▶ **Theorem 1.5.** *For every sufficiently large n and an integer $k = O(n^{2/3})$, there exists an n-vertex DAG G of diameter 4 and width $\omega = \Theta(k)$, such that every $(3, \epsilon = 1/2)$-shortcut H for G requires $\Omega(\omega^2)$ edges.*

Providing a tight lower bound as a function of ϵ, it is an interesting open problem.

Construction of (d, ϵ)-Shortcuts. We characterize the size and diameter tradeoff as a function the width, for any given ϵ. The algorithm for (d, ϵ)-shortcuts also serves the basis for our subsequent results.

▶ **Theorem 1.6.** *For every n-vertex ω-width, integer $d \geq 5$ and a slack parameter $\epsilon \in (0, 1)$, one can compute in $O(\omega \cdot m/\epsilon + m^{1+o(1)})$ time, a (d, ϵ)-shortcut with $\widetilde{O}(\omega^2/(\epsilon \cdot d) + n)$ edges. These shortcuts provide a distance bound d between each $u \in V$ to $(1 - \epsilon)$ fraction of its reachable vertices. The construction is randomized and the correctness holds w.h.p.*

Prior work of Kogan and Parter [18] provided constructions of d-shortcuts (without slack) whose size bounds are parameterized by the width. For example, they show a construction of d-shortcuts with $\widetilde{O}(n\omega/d^2 + n\sqrt{\omega}/d)$ edges. Our slack results of Thm. 1.6 improve over [18] mainly in the setting where $d = O(1)$. I.e., for $\omega = O(\sqrt{n})$, there is a near-linear $(5, \epsilon)$ shortcut, while the constructions of [18] require super-linear number of edges for any constant diameter. The construction of Theorem 1.6 allows one to be able to solve the $(1-\epsilon)$ single-source reachability problem (where we compute only $(1-\epsilon)$ fraction of the reachable vertices from a given source) in the parallel setting [20]. Given the computation of $(O(1), \epsilon)$ shortcut H, this problem can be solved in near-linear work (for graphs with width $O(\sqrt{n})$ and constant depth.

A closely related notion is *good-on-average shortcuts* which provide a small average directed distance in G. By combining our shortcuts with slack constructions with the constructions of standard d-shortcuts of [18], we have:

▶ **Theorem 1.7** (Good-on-Average Shortcuts). *For every n-vertex DAG G with width $\omega \le n^{2/5}$, one can compute a shortcut set of $\widetilde{O}(n)$ edges whose addition reduces the* average path length *to at most* 6.

Without a slack, the diameter upper bound for $n^{2/5}$-width graphs obtained with linear size shortcut is $n^{1/5}$ [18].

Separation between Slack vs. No Slack Shortcuts. We generalize the recent lower bound result of [28] to provide a lower bound graph with bounded width. For the regime of linear-size shortcuts, we show:

▶ **Theorem 1.8** (Separation). *There exists an n-vertex DAG G with width $\omega = \Theta(\sqrt{n}/\text{poly}\log n)$ that requires cn edges to reduce its diameter to $n^{1/6}/\text{poly}\log n$ for some constant $c > 0$. In contrast, $cn/2$ shortcut edges can reduce the distance to at most $O(1)$ for $(1-\epsilon)$ fraction of the $TC(G)$-pairs for some constant $\epsilon \in [0, 1]$.*

We observe that the lower bound graph of Huang and Pettie [13] also provides diameter bound of $\Omega(n^{1/6})$ for linear size shortcuts. The width bound of their construction is linear as written, and can be modified to provide a width bounded by $\Theta(n^{5/6})$[4]. By working with the lower bound of [28], we can reduce the width of their construction from $\Theta(n^{3/4})$ to $O(\sqrt{n})$.

1.1.2 Transitive Reduction and All-Successors Data Structures, with Slack

For a given DAG $G = (V, E)$ a transitive reduction is a graph $G' = (V, E')$ such that $TC(G) = TC(G')$ and $|E'| \le |E|$. We introduce the notion of *Transitive Reduction with Slack* in which the output graph G' satisfies that (i) $TC(G') \subseteq TC(G)$ and (ii) $|TC(G')| \ge (1-\epsilon)|TC(G)|$.

▶ **Theorem 1.9.** *For every n-vertex m-edge DAG $G = (V, E)$ with width ω and $\epsilon \in (0, 1)$, one can compute in time $\widetilde{O}(\omega m/\epsilon + m^{1+o(1)})$ an ϵ-slack transitive reduction graph $G' = (V, E')$ with $|E'| = \widetilde{O}(\omega^2/\epsilon + n)$ edges.*

[4] This seems to be the best that one can hope for in their construction, as their lower bound graph connects a collection of $\Theta(n)$ critical pairs by paths of length $\Omega(n^{1/6})$.

We complement this construction by a near matching lower bound that holds for any compression scheme of the transitive closure, with a constant slack parameter. By using connections to Error-Correcting-Codes, we show:

▶ **Theorem 1.10.** *There is an n-vertex DAG $G = (V, E)$ with width ω such that* any *data-structure that stores* 0.999*-fraction of the edges in* $TC(G)$ *requires space of* $\Omega(\omega^2 + n)$*.*

Near Optimal All-Successors Data-Structure, with Slack. We provide constructions of ϵ-slack all-successors data-structures. In this setting, it is required to compute a low-space data-structure that given a query node u, returns a list containing $(1-\epsilon)$ fraction all the successors of u in G. In a highly influential paper, [14] presented a near-optimal construction for the case of $\epsilon = 0$ (i.e., with no-slack). We provide an alternative construction for any slack parameter ϵ with near-optimal space and query time.

▶ **Theorem 1.11.** *For every n-vertex m-edge DAG G with width ω, one can compute in time $\widetilde{O}(\omega m/\epsilon + m^{1+o(1)})$ a data-structure of size $O(\omega^2/\epsilon + n)$ that given a vertex v can report a list L of at least $(1-\epsilon)$ fraction of all the successors (or predecessors) of v in time $\widetilde{O}(|L|)$.*

This improves upon the space bound of $(\omega \cdot n)$ by [14] for any $\omega = o(n)$. The space optimality follows by Theorem 1.10.

Handling General Graphs. Our constructions are parameterized by the *width* of the graph which is defined for DAGs (as in [18]). A plausible approach for handling general graphs G is by considering their corresponding DAG G' obtained by contracting each strongly-connected-component (SCC) in G into a node in G'. In the context of slack, this provides a node-weighted variant where the weight of each G'-node is given by the number of vertices in the SCC that it represents. On a high-level, our upper bound constructions might be adapted to this weighted setting at the cost of increasing the size bound by a factor of $O((\log n/\epsilon)^2)$.

2 Technical Overview

Throughout, it is convenient to consider a topological ordering $\{v_1, \ldots, v_n\}$ of the vertices V. We write $v_i \prec v_j$ if $i \leq j$. A chain is a dipath in the transitive closure $TC(G)$. For a chain $C = [v_1, \ldots, v_\ell]$, let $C[., v_i] = C[v_1, \ldots, v_i]$ and $C[v_i, .] = [v_i, \ldots, v_\ell]$. Furthermore we always assume that $G = (V, E)$ denote a DAG with vertex set V and edge set E, with $|V| = n$ and $|E| = m$. We focus here on the upper and lower bounds for shortcuts with slack, which capture most of the key intuition. Kogan and Parter [17] introduced a *path-centric* approach for computing reachability shortcuts. Their constructions of d-shortcuts with $\widetilde{O}(n^2/d^3)$ edges is based on connecting a collection of $\widetilde{O}(n/d)$ randomly selected **vertices** V' to a set of $\widetilde{O}(n/d^2)$ **paths** $\mathcal{P}'$, which are randomly sampled from a carefully chosen set $\mathcal{P}$ of vertex-disjoint $TC(G)$-dipaths. Their algorithm then adds one edge $e(v, P)$ between each vertex $v \in V'$ to each path $P \in \mathcal{P}'$. The edge $e(v, P)$ is chosen to be the edge that connects v to its first reachable vertex on P (note that such edge might not exist in which case we add nothing). The key observation in this context is that the edge $e(v, P) = (v, u)$ captures all reachability relations between v to the vertices on P, in particular, $z \in P$ is reachable from v iff $z \in P[u, .]$.

The key limitation for improving the tradeoff of [17] is that unlike vertex-path pairs, one cannot capture the reachability between pairs of paths by adding a single edge between some $v \in V(P_1)$ and $u \in V(P_2)$. (If this property would have held, then one can obtain linear d-shortcuts with $d = \widetilde{O}(n^{1/4})$, which it tight by [4].) Our key starting observation shows that by adding a *single* shortcut edge between pairs of paths P_i, P_j, one can capture the reachability relations for $1/\log n$ fraction of the vertex pairs in $V(P_i) \times V(P_j)$.

▶ **Observation 2.1.** *Let $P_1, P_2 \subseteq TC(G)$ be two dipaths with $|P_1|, |P_2| \leq n$. Then, there exists a single edge $e = (u, v) \in (V(P_1) \times V(P_2)) \cap TC(G)$ such that $G' = P_1 \cup \{(u, v)\} \cup P_2$ satisfies the following (i) $TC(G') \subseteq TC(G)$ and (ii) $|(V(P_1) \times V(P_2)) \cap TC(G')| = \Omega(|(V(P_1) \times V(P_2)) \cap TC(G)|/\log n)$.*

Proof. Let $P_1 = [u_1, \ldots, u_{\ell_1}]$ and $P_2 = [v_1, \ldots, v_{\ell_2}]$. For each $u_i \in P_1$, let a_i be the number of vertices that are reachable from u_i on P_2. Note that the vertex $z_i = v_{\ell_2 - a_i + 1}$ is then the first reachable vertex from u_i on P_2, as all the a_i vertices that are reachable from u_i are on the segment $P_2[z_i, .]$. We then have that $|(V(P_1) \times V(P_2)) \cap TC(G)| = \sum_{i=1}^{n} a_i$. By adding an edge $e_i = (u_i, z_i)$ to H', we provide a directed path for $i \cdot a_i$ pairs, namely from every $z \in P_1[u_1, u_i]$ to $z' \in P_2[z_i, .]$. Our goal is to show that $M = \max_i(a_i \cdot i) = \Omega\left(|TC(G) \cap V(P_1) \times V(P_2)|/\log n\right)$. That is, that there exists an edge e_i which reduces the distance for $1/\log n$ fraction of the pairs in $(V(P_1) \times V(P_2)) \cap TC(G)$. This holds as,

$$|(V(P_1) \times V(P_2)) \cap TC(G)| = \sum_{i=1}^{n} a_i \leq M/1 + M/2 + \ldots + M/n \leq O(M \cdot \log n) \ ,$$

hence, $M = \Omega(|(V(P_1) \times V(P_2)) \cap TC(G)|/\log n)$, as desired. ◀

We use this observation to provide $(5, \epsilon = 1/\log n)$-shortcuts. All our shortcut constructions start by computing the *minimum chain cover* (MCC) of G given by $\mathcal{C} = \{C_1, \ldots, C_\omega\}$. The MCC is minimum set of vertex-disjoint chains that cover all the vertices. Very recently, Cáceres provides an elegant almost-linear time algorithm for computing the MCC [5].

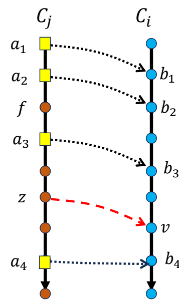

Figure 1 $(5, \epsilon)$ Shortcuts. Black dashed edges are in the shortcut, while the red edge is not.

Take I: 5-Shortcuts with a Logarithmic Slack. For every chain $C_i \in \mathcal{C}$, let H_i be a 2-shortcut for C_i of size $O(|C_i| \log n)$. See Lemma 3.2. Next, for each pair of chains C_j, C_i, let $e_{j,i}$ be the edge that satisfies the requirements of Obs. 2.1 (i.e., that maximizes the number of reachable pairs when adding $e_{j,i}$). The final shortcut is given by $H = \bigcup_i (H_i \cup C_i) \cup \bigcup_{j,i} \{e_{j,i}\}$. It is easy to see that $|H| = O(\omega^2 + n \log n)$. Consider the distance guarantee. By Obs. 2.1, for every pair C_j, C_i, by adding an edge $e_{j,i} = (x, y)$, there is a directed path in $G \cup \{e_{j,i}\}$ for a subset of pairs $S_{j,i} \subseteq V(C_j) \times V(C_i)$ such that $|S_{j,i}| = \Omega(|(V(C_j) \times V(C_i)) \cap TC(G)|/\log n)$. For every $(a, b) \in S_{j,i}$, the dipath is given by $C_j[a, x] \circ e_{j,i} \circ C_i[y, b]$. By adding the 2-shortcuts of H_i and H_j, we get that $\text{dist}_{G \cup H_j}(a, x) \leq 2$ and $\text{dist}_{G \cup H_i}(y, b) \leq 2$, hence $\text{dist}_{G \cup H}(a, b) \leq 5$.

Take II: 5-Shortcuts for any Slack. To provide $(5, \epsilon)$-shortcuts for any slack parameter ϵ, we will be adding a collection of $O(\log n/\epsilon)$ edges between each pair of chains C_j, C_i. This will provide a distance bound of 5 from each v to a $(1 - \epsilon)$-fraction of its incoming $TC(G)$-neighbors. Fix chains $C_j, C_i \in \mathcal{C}$. We mark $\ell = O(\log n/\epsilon)$ vertices on $C_j = [v_1, \ldots, v_k]$. The

q^{th} marked vertex is v_{s_q} where $s_q = \lceil (1+\epsilon)^q \rceil$. Let $A_j = \{a_1, \ldots, a_\ell\}$ be the collection of marked vertices according to their appearance on C_j. For every $a_q \in A_j$, let b_q be the first reachable vertex from a_q on C_i.

The algorithm then adds the edges $H_{j,i} = \{(a_q, b_q) \mid a_q \in A_j\}$. The final shortcut is given by $H = \bigcup_i (H_i \cup C_i) \cup \bigcup_{j,i} H_{j,i}$, where H_i is a 2-shortcut for C_i, as in the previous construction. The size bound is again immediate and we focus on the distance guarantee. Fix a vertex $v \in C_i$ and a chain C_j. Let $N_j(v) = \{z \in C_j \mid (z,v) \in TC(G)\}$ be the set of all the incoming $TC(G)$-neighbors of v in C_i. We claim that there exists a subset $I(v) \subseteq N_j(v)$ such that $|I(v)| \geq (1-2\epsilon)|N_j(v)|$ and that $\text{dist}_{G \cup H}(u,v) \leq 5$ for every $u \in I(v)$. As this would hold for every C_j and every v, the slack guarantees follows. Let z be the lowest (i.e., last) vertex on C_j that belongs to the set $N_j(v)$.

Let a, a' be the closest marked vertices to z on C_j (in the figure, $a = a_3, a' = a_4$), where $a \prec z \prec a'$, and let b be the first reachable vertex from a on C_i. Since $(a,v) \in TC(G)$, we have that $b \prec v$. We set $I(v) = C_j[., a]$. For every $f \in I(v)$, we have the following directed f-v path: $C_j[f,a] \circ (a,b) \circ C_i[b,v]$. Since $H_i, H_j \subseteq H$, we have that $\text{dist}_{G \cup H}(f,v) \leq 5$ for every $f \in I(v)$. Finally, we bound the size of $I(v)$. By definition, $N_j(v) = V(C_j[.,z])$. By the definition of the marked vertices a, a' we have that $|C_j[a,a']| \leq \epsilon |I(v)|$. Hence, $|I(v)| \geq |N_j(v)|/(1+\epsilon) \geq (1-2\epsilon)|N_j(v)|$.

Lower Bounds for $(3,\epsilon)$ Shortcuts. Many of the current lower-bound constructions of linear size shortcuts[5] (with no slack) are based on defining a set of $\Theta(n)$ critical pairs C, with some desirable disjointness properties of their respective shortest paths, and claiming that every shortcut edge can reduce the distance between a constant number of pairs, leading to a lower bound of $\Omega(|C|)$. In the slack model, these constructions would not provide any meaningful result, as one can simply neglect all critical pairs, which constitutes only $|C|/n^2 = 1/n$ fraction of the total edges in the transitive closure. The slack model, say with $\epsilon = 1/2$, calls then for new lower bound constructions in which the number of critical pairs is sufficiently *large*, i.e., at least some *constant* fraction of the total size of the transitive closure.

We present two lower bound constructions for $(3,\epsilon)$ shortcuts. The first holds for any width value ω but restricted to non-constant value of ϵ where $\epsilon = 2^{-\sqrt{\log \omega}}$. The second construction holds for any constant value of ϵ but restricted to graphs with width of $O(n^{2/3})$. Both constructions are based on a 4-diameter layered graph with 5 layers, $L_0, \ldots, L_4$. The key difference is in the bipartite graph that we embed between the internal layers. Our first construction (Theorem 1.4) connects layers L_2, L_3 by embedding the Ruzsa-Szemerédi graph which, informally, contains an almost quadratic number of edges that can be partitioned into a linear number of induced matchings. The second construction (Theorem 1.5) embeds an Erdős-Rényi graph between layers L_1, L_2 and L_3. In what follows, we sketch the high-level structure of the first construction which is based on the existence of a Ruzsa-Szemerédi (RS) graph from [24]. Formally, a graph is (r,t)-RS-digraph if its edges can partitioned into t pairwise disjoint induced matchings, each of size r. Ruzsa and Szemerédi in [24] provide a construction of n-vertex (r,t) RS graphs with $t = \Theta(n)$ and $r = n/2^{\sqrt{\log n}}$. We draw the a connection between the achievable r, t bounds and the size of $(3,\epsilon)$ shortcuts.

▶ **Lemma 2.2.** *[From RS to LB for Shortcuts with Slack] Given an (r,t) RS k-vertex graph, then for every $n \geq c \cdot k$ for some constant $c \geq 1$, there is an n-vertex graph with width $\omega = \Theta(k)$ for which any $(3, \epsilon = \Theta(r/\omega))$ shortcut requires $\Omega(t \cdot r)$ edges.*

[5] An exception is the very recent lower bound of Bodwin and Hoppenworth [4].

The graph has 5 layers L_0, L_1, L_2, L_3, L_4 where the first and last layers, L_0, L_4 consists of a collection of $k = \Theta(\omega)$ vertex-disjoint directed paths each of length $N = \lceil n/k \rceil$ and the RS graph is used to connect layer L_2 with L_3. The size of the layers is given by $|L_0|, |L_4| = \Theta(n)$, $|L_1| = t$ and $|L_2| + |L_3| = k$. We connect all vertices of the i^{th} path in layer 1 as incoming vertices of the i^{th} vertex in layer 2. Similarly, the i^{th} vertex on layer L_3 has outgoing edges to all the vertices on the i^{th} path of layer L_4. Importantly, the j^{th} vertex on layer L_1 is connected to the L_2-endpoints of the j^{th} induced matching in the RS graph connecting L_2 and L_3. Our argument shows that there is a collection of $a = n^2 \cdot r/k$ *critical* pairs C with some useful disjointness properties. Since $\epsilon = a/n^2 = \Theta(r/k)$, a $(3, \epsilon)$ shortcut must provide a distance 3 for at least a constant fraction of the pairs in C. Using the disjointness properties on C, we then claim that each shortcut edge can reduce the distance for at most $b = O(n^2/(tk))$ pairs. Hence, one needs to add a $\Omega(a/b) = \Omega(t \cdot r)$ shortcut edges.

2.1 Preliminaries

Graph Notations. For an n-vertex digraph $G = (V, E)$, let $TC(G)$ denote its transitive closure. Throughout, let $n = |V|$ and $m = |E|$. For an a-b dipath P and an b-c dipath P' the concatenation of the paths is denoted by $P \circ P'$. Let $|P|$ denote the number of vertices on P (unless mentioned otherwise). For a collection of paths $\mathcal{P}$, let $V(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} V(P)$. For an element set X and $p \in [0, 1]$, let $X[p]$ be the set obtained by taking each element of X into $X[p]$ independently with probability p.

For a chain $C = [v_1, \ldots, v_\ell]$, let $C(i) = v_i$, namely, the i^{th} vertex on the chain. Let $C[v_i, .]$ denote the sub-chain $[v_i, \ldots, v_\ell]$ and similarly, $C[., v_i] = [v_1, \ldots, v_i]$. For an integer $k \in \{1, \ldots, \ell\}$, the k-length prefix (resp., suffix) of C are given by $\mathsf{Pre}(C_i, k) = [v_1, \ldots, v_k]$ and $\mathsf{Suff}(C_i, k) = [v_{\ell-k+1}, \ldots, v_\ell]$. For v_i, v_j with $i \le j$, let $C[v_i, v_j] = [v_i, \ldots, v_j]$, $C[v_i, v_j) = [v_i, \ldots, v_{j-1}]$ and $C(v_i, v_j] = [v_{i+1}, \ldots, v_j]$.

For a vertex u and a chain $C = [v_1, \ldots, v_\ell]$, let $\mathsf{Reach}(u, C)$ be the set of vertices on C that are reachable from u in G. Similarly, let $\mathsf{IN}(u, C)$ be the set of vertices $v \in C$ that have a directed path into u. Formally,

$$\mathsf{Reach}(u, C) = \{v \in C \mid (u, v) \in TC(G)\} \text{ and } \mathsf{IN}(u, C) = \{v \in C \mid (v, u) \in TC(G)\}\,. \quad (2.1)$$

Let $\mathsf{IN}(u) = \{v \in V(G) \mid (v, u) \in TC(G)\}\}$ be the set of all incoming neighbors of u in $TC(G)$. Denote $\mathsf{FirstReach}(u, C)$ as the the first (i.e., upmost on C) reachable vertex from u on $C = [v_1, \ldots, v_\ell]$, letting $i \in \{1, \ldots, \ell\}$ be the smallest index such that $v_i \in \mathsf{Reach}(u, C)$, then $\mathsf{FirstReach}(u, C) = v_i$. For a given subset $Z \subseteq V(C)$, let $\mathsf{FirstReach}(u, C, Z)$ be the upmost vertex in Z on C that is reachable from u. For a subset of edges H, let $H^R = \{(u, v) \mid (v, u) \in H\}$ be the edge set in which all H-edges are reversed.

Definitions of Shortcuts and TC Compression, with Slack. For a given n-vertex DAG $G = (V, E)$, an integer d and $\epsilon \in (0, 1)$, a (d, ϵ)-shortcut $H \subseteq TC(G)$ satisfies that there exists $E' \subseteq TC(G)$ such that $|E'| \ge (1 - \epsilon)|TC(G)|$ and $\text{dist}_{G \cup H}(u, v) \le d$ for every $(u, v) \in E'$. A *TC-compression* is a data-structure DS that upon a query (u, v) returns $DS((u, v)) = 1$ iff $(u, v) \in TC(G)$. A TC-compression with slack ϵ, DS_ϵ, satisfies the following: Let $E' = \{(u, v) \in V \times V \mid DS_\epsilon((u, v)) = 1\}$. Then, $E' \subseteq TC(G)$ and $|E'| \ge (1 - \epsilon) \cdot |TC(G)|$. Our constructions of (d, ϵ) shortcuts and TC-compression schemes satisfy a stronger slack guarantee: For any vertex u, there is a subset $\mathsf{O}(u) \subseteq \mathsf{Reach}(u)$ such that $|\mathsf{O}(u)| \ge (1 - \epsilon)|\mathsf{Reach}(u)|$ and the structure satisfy the distance (resp., reachability) guarantee for all pairs in $\{u\} \times \mathsf{Reach}(u)$. Same holds for the predecessor lists.

Due to space limitations, we only provide the constructions of shortcuts and TC compressions.

3 Shortcuts with Slack

3.1 Upper Bounds

The starting point to all of our constructions is the computation of the minimum chain cover $\mathcal{C} = \{C_1, \ldots, C_\omega\}$ of G. As observed by Dilworth and Fulkerson [9, 11], a MCC can be computed in polynomial time for DAGs, while being NP-hard for general graphs. Recently, an almost-linear time MCC algorithm was shown by [5].

We need the following definitions. For every vertex v, let $c(v)$ be the index i of the chain C_i such that $v \in V(C_i)$, and let $p(v)$ be the index of the position of v on C_i, i.e., v is the $p(v)^{th}$ vertex on C_i. Our (d, ϵ)-shortcuts are based on connecting a collection of $O(\log n/\epsilon)$ vertices that lie on exponentially growing positions $(1+\epsilon)^i$ on the chains $C_i \in \mathcal{C}$. For a given slack parameter $\epsilon \in (0, 1)$, let $\mathsf{ExpU}(C, \epsilon) = \{u_1, \ldots, u_k\} \subseteq C$ where[6] $p(u_1) = 1$ and for every $i \in \{1, \ldots, k-1\}$, we have $p(u_{i+1}) = p(u_i) + \lceil \epsilon p(u_i) \rceil$. Similarly, the set $\mathsf{ExpD}(C, \epsilon)$ corresponds to an exponentially spaced vertices starting from the bottom of the chain C, i.e., $\mathsf{ExpD}(C, \epsilon) = \mathsf{ExpU}(C^R, \epsilon)$, where C^R is the chain C with reversed directions.

▶ **Observation 3.1.** *Let $C = [v_1, \ldots, v_\ell]$ and C' be vertex disjoint chains. Then,*

$$|\mathsf{Reach}(v_1, C')| \geq |\mathsf{Reach}(v_2, C')| \geq \ldots \geq |\mathsf{Reach}(v_\ell, C')| \ .$$

▶ **Lemma 3.2** ([23] Lemma 1.1)**.** *For any dipath P, there is an algorithm* ShortcutPath *that computes in near linear time a $(2, \epsilon = 0)$-shortcut H with $|H| = O(|P| \log |P|)$ edges.*

In the full paper, we describe a construction of $(2, \epsilon = 1/2)$-shortcuts with $O(\omega n)$ edges, which is also tight. The construction of $(3, \epsilon = 1/2)$-shortcuts with $\widetilde{O}((\omega^2 + n)/\epsilon)$ edges is also in the full paper. On a high level, in this construction, for every pair of chains C_j and C_i, we add $|\mathsf{ExpU}(C, \epsilon)|$ shortcut edges between each $u \in \mathsf{ExpU}(C_j, \epsilon)$ to a unique $v \in \mathsf{ExpD}(C_i, \epsilon)$. We also connect each vertex in C_ℓ to all its $TC(G)$-neighbors in $\mathsf{ExpU}(C_\ell, \epsilon), \mathsf{ExpD}(C_\ell, \epsilon)$. These edges provide 1-hop distance to the marked vertices on the designated chains, which is critical to the diameter bound. Since we do not connect $u \in \mathsf{ExpU}(C_j, \epsilon)$ to its first reachable vertex on C_i, but rather to a marked vertex in $\mathsf{ExpD}(C_i, \epsilon)$, leads to a more delicate analysis.

We next prove Theorem 1.6 by providing a construction of a (d, ϵ)-shortcut H for any given diameter d and slack parameter ϵ. Our (d, ϵ)-shortcut in-fact satisfies a stronger slackness property.

▶ **Definition 3.3** (In-and-Out Covering)**.** *For a given n-vertex DAG G, $\epsilon \in (0, 1)$ and integer d, a vertex v is (d, ϵ) IN-covered by a shortcut set H if there exists a subset $I(v) \subseteq \mathsf{IN}(v)$ of cardinality at least $(1 - 2\epsilon)|\mathsf{IN}(v)|$ such that $\mathrm{dist}_{G \cup H}(u, v) \leq d$, for every $u \in I(v)$. Similarly, a vertex v is (d, ϵ) OUT-covered by H if there exists a subset $\mathsf{O}(v) \subseteq \mathsf{Reach}(v)$ of cardinality at least $(1 - 2\epsilon)|\mathsf{Reach}(v)|$ such that $\mathrm{dist}_{G \cup H}(v, u) \leq d$, for every $u \in \mathsf{O}(v)$.*

In the full version, we show:

▶ **Observation 3.4.** *Suppose that there is an algorithm that given an n-vertex ω-width graph G, ϵ, d computes a shortcut set H such that every vertex $v \in V$ is (d, ϵ) IN-covered by H and such that $|H| = g(n, \omega, \epsilon, d)$ for some function g. Then, there is an algorithm that computes a shortcut set H' such that every vertex v is (d, ϵ) OUT-covered by H' and $|H'| = g(n, \omega, \epsilon, d)$.*

[6] We define the vertex u_{i+1} in $\mathsf{ExpU}(C, \epsilon)$ inductively, rather than by $(1+\epsilon)^{i+1}$ for minor technical simplification of our analysis. Specifically, this choice guarantees that all k vertices are disjoint.

From the point on, we focus on computing a shortcut set H such that each $v \in V$ is (d,ϵ) IN-covered by H. Let $\mathcal{C}$ be a minimum chain cover for G and let $\mathcal{R} \subseteq \mathcal{C}$ be a sample of $O(\omega \log n/d)$ chains obtained by sampling each chain $C_i \in \mathcal{C}$ into $\mathcal{R}$ independently with probability of $p = \Theta(\log n/d)$. The algorithm adds to H a 2-shortcut set $\mathsf{ShortcutPath}(C_i, 2)$ for each $C_i \in \mathcal{C}$. Next, for each pair of chains $C_j, R_i \in \mathcal{C} \times \mathcal{R}$, the algorithm connects each $u \in \mathsf{ExpU}(C_j, \epsilon)$ to its first reachable vertex on R_i, $\mathsf{FirstReach}(u, R_i)$. This completes the description of the algorithm.

Algorithm $\mathsf{dSlackShortcutIN}(G, d, \epsilon)$**:**

Input: An n-vertex DAG $G = (V, E)$, slack parameter $\epsilon \in (0, 1]$, desired diameter d.
Output: $H \subseteq TC(G)$ such that each $v \in V$ is (d, ϵ) *IN-covered* by H.

1. Let $\mathcal{C} = \{C_1, \ldots, C_\omega\}$ be the minimum chain cover of G.
2. $\mathcal{R} = \mathcal{C}[p]$ for $p = \Theta(\log n/d)$.
3. $\forall C_i \in \mathcal{C}$, $H_i \leftarrow C_i \cup \mathsf{ShortcutPath}(C_i, 2)$.
4. For each $C_j \times R_i \in \mathcal{C} \times \mathcal{R}$ do the following:
   a. $H_{j,i} = \{(u, \mathsf{FirstReach}(u, R_i)\}_{u \in \mathsf{ExpU}(C_j, \epsilon)}$.
5. $H \leftarrow \bigcup_{i=1}^{\omega} H_i \cup \bigcup_{j,i} H_{j,i}$.

Algorithm $\mathsf{dSlackShortcut}(G, \epsilon)$**:**

Input: An n-vertex DAG $G = (V, E)$, slack parameter $\epsilon \in (0, 1]$, desired diameter d.
Output: $(d, 2\epsilon)$-shortcut $H \subseteq TC(G)$.

1. $H_{\mathsf{IN}} = \mathsf{dSlackShortcutIN}(G, d, \epsilon)$.
2. $H_{\mathsf{OUT}} = \mathsf{dSlackShortcutIN}(G^R = (V, E^R), d, \epsilon)$.
3. $H = H_{\mathsf{IN}} \cup H^R_{\mathsf{OUT}}$.

Correctness. We turn to analyze the construction and prove Theorem 1.6. The size bound is immediate as $|H_i| = O(|C_i| \log n)$, $|H_{j,i}| = O(\log n/\epsilon)$. The bound follows by summing over all vertex-disjoint chains in $\mathcal{C}$, and over the $\omega \cdot |\mathcal{R}| = O(\omega \log n/d)$ pairs of chains, where the last equality follows w.h.p. by Chernoff. The computation time is dominated by the (almost-linear) computation of the MCC and single-source reachability w.r.t $\widetilde{O}(\omega/\epsilon)$ sources. Hence, the time bound is $\widetilde{O}(\omega \cdot m/\epsilon + m^{1+o(1)})$. We use this definition:

▶ **Definition 3.5.** *Given a vertex v, a chain $C \in \mathcal{C}$ and a shortcut set H, we say that v is (d, ϵ, C)-covered by H if there exists a subset $Z \subseteq \mathsf{IN}(v, C)$ such that: (i) $|Z| \geq (1 - 2\epsilon)|\mathsf{IN}(v, C)|$ and (ii) $\text{dist}_{G \cup H}(u, v) \leq d$ for every $u \in Z$. A vertex v is (d, ϵ)-covered by H if v is (d, ϵ, C_j)-covered by H, for every chain C_j in the minimum chain cover $\mathcal{C}$.*

▶ **Lemma 3.6.** *For every $C_j \in \mathcal{C}$ and every vertex $v \in V(\mathcal{R})$, v is $(5, \epsilon, C_j)$ IN-covered by H.*

Proof. Let $R_i \in \mathcal{R}$ be such that $v \in R_i$, and let u be the last (downmost) vertex on C_j that appears in $\mathsf{IN}(v, C_j)$. Also, let a, b be the closest vertices in $\mathsf{ExpU}(C_j, \epsilon)$ that appear above (respectively below) u on C_j. By the definition of $\mathsf{ExpU}(C_j, \epsilon)$, it holds that $|C_j[a, b]| \leq \epsilon |C_j[., a]|$. Since $b \notin \mathsf{IN}(v, C_j)$, we have that $|C_j[., a]| \geq (1 - 2\epsilon)|\mathsf{IN}(v, C_j)|$. We next show that $\text{dist}_{G \cup H}(z, v) \leq d$ for every $z \in C_j[., a]$. Let $z' = \mathsf{FirstReach}(a, R_i)$, then $(z', a) \in H_{j,i}$ and since z' is not below v on C_j, we have: $\text{dist}_{G \cup H}(z, v) \leq \text{dist}_{G \cup H}(z, a) + 1 + \text{dist}_{G \cup H}(z', v) \leq 5$, where $\text{dist}_{G \cup H}(z, a), \text{dist}_{G \cup H}(z', v) \leq 2$ by the addition of the 2-shortcut sets of C_j and R_i. ◀

We next claim that every vertex $v \in V$ is (d, ϵ) IN-covered in H. This provides an incoming distance d bound into v from $(1-\epsilon)$ fraction of its incoming $TC(G)$-neighbors.

▶ **Lemma 3.7.** *W.h.p., any vertex $v \in V$ in (d, ϵ) IN-covered by H.*

Proof. Fix $C_j \in \mathcal{C}$ and let u be the last vertex in C_j that appears in $\mathsf{IN}(v, C_j)$. Let $P_{u,v}$ be u-v shortest path in $G' = G \cup \bigcup_{C_i \in \mathcal{C}} H_i$. If $|P_{u,v}| \leq d-2$, then we are done as for every $z \leq u$, we have $\text{dist}_{G \cup H}(z, v) = \text{dist}_{G \cup H}(z, u) + \text{dist}_{G \cup H}(u, v) \leq d$.
Assume that $|P_{u,v}| \geq d-1$. We next claim that for every $C_i \in \mathcal{C}$, it holds that

$$|P_{u,v} \cap V(C_i)| \leq 3 \ . \tag{3.1}$$

This holds since by including the 2-shortcut set H_i of C_i and as G is a DAG, it holds that $\text{dist}_{G'}(x, y) \leq 2$ for every $x, y \in C_i$. Since $P_{u,v}$ is a shortest path in G', we have that it can contain at most 3 vertices from each C_i.

Consider the $d/2$-length suffix of $P_{u,v}$, denote this segment by P'. We then claim that w.h.p. $V(P') \cap V(\mathcal{R}) \neq \emptyset$. This holds since by Eq. (3.1), P' has at most 3 vertices from each $C_i \in \mathcal{C}$, and consequently it intersects with $\Omega(d)$ distinct chains in $\mathcal{C}$. Since each chain C_i is sampled into $\mathcal{R}$ with probability of $p = \Theta(\log n/d)$, we have that w.h.p. there exists $R_i \in \mathcal{R}$ such that there exists $z \in V(R_i) \cap V(P') \neq \emptyset$.

Since $z \in P_{u,v}$, then $\mathsf{IN}(z, C_j) \subseteq \mathsf{IN}(v, C_j)$. However, by the selection of u (i.e., downmost vertex on C_j that appears in $\mathsf{IN}(v, C_j)$), it holds that $\mathsf{IN}(z, C_j) = \mathsf{IN}(v, C_j)$. Since $z \in V(\mathcal{R})$, by Lemma 3.6 we have that z is $(5, \epsilon, C_j)$-covered by H. As $\text{dist}_{G \cup H}(z, v) \leq d/2$, we have that v is $(5 + d/2, \epsilon, C_j)$-covered by H. To see this, let $Z' \subseteq \mathsf{IN}(z, C_j)$ be such that $|Z'| \geq (1-\epsilon)|\mathsf{IN}(z, C_j)|$ and $\text{dist}_{G \cup H}(x, z) \leq 5$ for every $x \in Z'$. Then, $\text{dist}_{G \cup H}(x, v) \leq 5 + d/2$ for every $x \in Z'$ and the claim holds as $\mathsf{IN}(v, C_j) = \mathsf{IN}(z, C_j)$. ◀

4 Transitive Closure Compression and All-Successors Data-Structures

4.1 Upper Bounds

In this section, we employ our construction scheme for (d, ϵ)-shortcuts with $d = O(1)$ to provide several compression schemes of the transitive closure, up to an ϵ slack.

Transitive Reduction with Slack. For a given DAG $G = (V, E)$ a transitive reduction is a graph $G' = (V, E')$ such that $TC(G) = TC(G')$ and $|E'| \leq |E|$. We introduce the notion of ϵ-slack transitive reduction graph in which the output graph G' satisfies that (i) $TC(G') \subseteq TC(G)$ and (ii) $|TC(G')| \geq (1-\epsilon)|TC(G)|$. We next turn to prove Thm. 1.9.

The compression algorithm. For simplicity, we present an algorithm that preserves $(1-\epsilon)$-fraction of the incoming $TC(G)$-neighbors of v for each $v \in V$. To handle the outgoing $TC(G)$-edges of each $v \in V$, employ the algorithm on the graph G^R and then reverse the edges of the output graph. For a graph G', let $\mathsf{IN}_{TC(G')}(u) = \{v \mid (v, u) \in TC(G')\}$. Given a graph G and $\epsilon \in (0, 1)$, we compute a graph $G_{\mathsf{IN}} = (V, E_{\mathsf{IN}})$, as follows. For every pair of chains $C_j, C_i \in \mathcal{C}$ and every marked vertex $u \in \mathsf{ExpU}(C_j, \epsilon)$, the algorithm adds to E_{IN}, the edge $(u, \mathsf{FirstReach}(u, C_i))$. The analysis of the construction is deferred to the full version.

All-Successors Data-Structure with Slack. Following Jagadish [14], we call the list of all vertices reachable from a vertex u, the *successor list* of u. Similarly, we the list of all vertices that have an incoming path into u, the *predecessor list* of u.

In what follows, we the approximate predecessor list of the vertices. The approximate successor list can obtained by working on the reversed graph G^R.

Data-Structure Construction. Let $\mathcal{C} = \{C_1, \ldots, C_\omega\}$ be a minimum chain cover of G. For each $v \in V$, we store the values $(c(v), p(v))$ where $c(v)$ is the index of the chain to which v belongs, and $p(v)$ is the index of v on that chain. That is, v is the $p(v)^{th}$ vertex on the chain $C_{c(v)}$. For each chain $C_i \in \mathcal{C}$, we store the following $O(\omega \log n/\epsilon)$ bits of information:

- For every $C_j \in \mathcal{C}$, let $a_{j,i}$ be the *index* of the upmost vertex on C_i that has an incoming path from some vertex $v' \in C_j$. Let $S_i = [(a_{j_1,i}, j_1), \ldots, (a_{j_\omega,i}, j_\omega)]$ be a list sorted by the value of $a_{j,i}$, where $a_{j_1,i} \leq a_{j_2,i} \leq \ldots \leq a_{j_\omega,i}$.
- For every $C_j \in \mathcal{C}$ let $\mathsf{ExpU}(C_j, \epsilon) = \{u_{j,1}, \ldots, u_{j,k}\}$ where $u_1 \prec \ldots \prec u_k$. For every $u_{j,\ell}$, let $v_{j,\ell} = \mathsf{FirstReach}(u, C_i)$. Let $Q_{i,j} = [(u_{j,1}, v_{j,1}), \ldots, (u_{j,k}, v_{j,k})]$. (Note that by Obs. 3.1, we also have that $v_{j,1} \prec \ldots \prec v_{j,k}$.)

The data-structure $DS_{\mathsf{IN}}(G, \epsilon)$ consists of the following items:

- $\{(c(v), p(v)\}_{v \in V}$
- S_i, $i \in \{1, \ldots, \omega\}$.
- $Q_{i,j}$, $i, j \in \{1, \ldots, \omega\}$.

The final data-structure is given by the union of $DS_{\mathsf{IN}}(G, \epsilon)$ and $DS_{\mathsf{IN}}(G^R, \epsilon)$.

Query Algorithm. Given a vertex v, we first extract $c(v), p(v)$ in $O(1)$ time. Let $c(v) = i$. Set parameter ℓ, as follows. If $a_{j_\omega,i} \leq p(v)$, $\ell = \omega + 1$. Otherwise, let ℓ be such that $a_{j_\ell,i} > p(v)$ which can be found by Binary search on S_i. The algorithm iterates over each $C_j \in \{C_{j_1}, \ldots, C_{j_{\ell-1}}\}$. For each such C_j, it holds that v has some $TC(G)$-incoming neighbors from $V(C_j)$ and the algorithm will output a (possibly partial) non-empty list of all these neighbors, by using the list $Q_{i,j}$, as follows. Apply a Binary search on $Q_{i,j}$ to detect the deepest vertex $u \in \mathsf{ExpU}(C_j, \epsilon)$ such that $\mathsf{FirstReach}(u, C_i)$ is not below v on C_i. The algorithm then outputs $I(v) = V(C_j[., u])$. Theorem 1.11 follows by the following Lemmas.

▶ **Lemma 4.1.** *The total space of the data-structure is $O(\omega^2 \log n/\epsilon + n \log n)$ and the construction time is $\widetilde{O}((\omega \cdot m)/\epsilon + m^{1+o(1)})$.*

▶ **Lemma 4.2.** *For every v, the algorithm returns a list $I(v)$ that consists of at least $(1-\epsilon)$-fraction of the predecessor list, $\mathsf{IN}_{TC(G)}(v)$, of v. Moreover, the query time is $\widetilde{O}(|I(v)|)$.*

Proof. Let $i = c(v)$ and fix $C_j \in \mathcal{C}_i$. We claim that the algorithm outputs $(1-\epsilon)$ fraction of the predecessors of v on C_j. Let z be the lowest vertex on C_j that appears in the predecessor list of v. Let u, u' be the closest marked vertices to z in $\mathsf{ExpU}(C_j, \epsilon)$ that appear below and above z on C_j, respectively. The algorithm then outputs all vertices on $C_j[., u]$. Since no vertex on $C_j[z, .]$ is a predecessor of v, and as $|C_j[u, u']| \leq \epsilon |C_j[., u]|$, we have that $|C_j[., u]| \geq (1-\epsilon)|C_j[., z]|$. The claim follows. We next analyze the query time. The indices $c(v), p(v)$ can be computed in $O(1)$ as we explicitly store it, for each vertex v. By applying a Binary search on S_i, the algorithm inspects only chains C_j such that $V(C_j) \cap \mathsf{IN}_{TC(G)}(v) \neq \emptyset$. Next for each such chain, detecting the lowest marked vertex on C_j that belongs to $\mathsf{ExpU}(C_j, \epsilon)$ that has an incoming path to v can be done in $O(\log \log n)$ time (for fixed ϵ). This holds since the list $Q_{i,j}$ is sorted by the depth of the first reachable vertex $\mathsf{FirstReach}(u, C_i)$ for every $u \in \mathsf{ExpU}(C_j, \epsilon)$. ◀

References

1 Ittai Abraham, Yair Bartal, T.-H. Hubert Chan, Kedar Dhamdhere, Anupam Gupta, Jon M. Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 83–100. IEEE Computer Society, 2005.

2 Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 271–286. ACM, 2006.

3 Rakesh Agrawal. Alpha: An extension of relational algebra to express a class of recursive queries. *IEEE Transactions on Software Engineering*, 14(7):879–885, 1988.

4 Greg Bodwin and Gary Hoppenworth. Folklore sampling is optimal for exact hopsets: Confirming the $\sqrt{n}$ barrier. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 701–720. IEEE, 2023.

5 Manuel Cáceres. Minimum chain cover in almost linear time. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 31:1–31:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

6 T.-H. Hubert Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In Yossi Azar and Thomas Erlebach, editors, *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, volume 4168 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2006.

7 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000.

8 Shaul Dar. *Augmenting databases with generalized transitive closure*. University of Wisconsin at Madison, 1993.

9 RP Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, pages 161–166, 1950.

10 Michael Dinitz. Compact routing with slack. In Indranil Gupta and Roger Wattenhofer, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, Portland, Oregon, USA, August 12-15, 2007*, pages 81–88. ACM, 2007.

11 Delbert R Fulkerson. Note on dilworth's decomposition theorem for partially ordered sets. In *Proc. Amer. Math. Soc*, volume 7, pages 701–702, 1956.

12 William Hesse. Directed graphs requiring large numbers of shortcuts. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA*, pages 665–669. ACM/SIAM, 2003.

13 Shang-En Huang and Seth Pettie. Lower bounds on sparse spanners, emulators, and diameter-reducing shortcuts. In David Eppstein, editor, *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden*, volume 101 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

14 H. V. Jagadish. A compression technique to materialize transitive closure. *ACM Trans. Database Syst.*, 15(4):558–598, 1990.

15 Wojciech Jaśkowski and Krzysztof Krawiec. Formal analysis, hardness, and algorithms for extracting internal structure of test-based problems. *Evolutionary computation*, 19(4):639–671, 2011.

16 Jon M. Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 444–453. IEEE Computer Society, 2004.

17 Shimon Kogan and Merav Parter. New diameter-reducing shortcuts and directed hopsets: Breaking the barrier. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1326–1341. SIAM, 2022.

18 Shimon Kogan and Merav Parter. Faster and unified algorithms for diameter reducing shortcuts and minimum chain covers. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 212–239. SIAM, 2023.

19 Shimon Kogan and Merav Parter. Towards bypassing lower bounds for graph shortcuts. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 73:1–73:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

20 Yang P. Liu, Arun Jambulapati, and Aaron Sidford. Parallel reachability in almost linear work and square root depth. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1664–1686. IEEE Computer Society, 2019.

21 Kevin Lu, Virginia Vassilevska Williams, Nicole Wein, and Zixuan Xu. Better lower bounds for shortcut sets and additive spanners via an improved alternation product. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3311–3331. SIAM, 2022.

22 David Peleg. Low stretch spanning trees. In Krzysztof Diks and Wojciech Rytter, editors, *Mathematical Foundations of Computer Science 2002, 27th International Symposium, MFCS 2002, Warsaw, Poland, August 26-30, 2002, Proceedings*, volume 2420 of *Lecture Notes in Computer Science*, pages 68–80. Springer, 2002.

23 Sofya Raskhodnikova. Transitive-closure spanners: A survey. In Oded Goldreich, editor, *Property Testing - Current Research and Surveys*, volume 6390 of *Lecture Notes in Computer Science*, pages 167–196. Springer, 2010. `doi:10.1007/978-3-642-16367-8_10`.

24 Imre Z Ruzsa and Endre Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18(939-945):2, 1978.

25 Mikkel Thorup. On shortcutting digraphs. In Ernst W. Mayr, editor, *Graph-Theoretic Concepts in Computer Science, 18th International Workshop, WG '92, Wiesbaden-Naurod, Germany, June 19-20, 1992, Proceedings*, volume 657 of *Lecture Notes in Computer Science*, pages 205–211. Springer, 1992.

26 David S Touretzky. *The mathematics of inheritance systems*, volume 8. Morgan Kaufmann, 1986.

27 Jeffrey D. Ullman and Mihalis Yannakakis. High-probability parallel transitive-closure algorithms. *SIAM J. Comput.*, 20(1):100–125, 1991.

28 Virginia Vassilevska Williams, Yinzhan Xu, and Zixuan Xu. Simpler and higher lower bounds for shortcut sets. *SODA*, abs/2310.12051, 2024. `doi:10.48550/arXiv.2310.12051`.

29 Mihalis Yannakakis. Graph-theoretic methods in database theory. In Daniel J. Rosenkrantz and Yehoshua Sagiv, editors, *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, April 2-4, 1990, Nashville, Tennessee, USA*, pages 230–242. ACM Press, 1990.

30 Mihalis Yannakakis. Perspectives on database theory. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 224–246. IEEE Computer Society, 1995.