

Approximation Algorithms and Hardness for n -Pairs Shortest Paths and All-Nodes Shortest Cycles

Mina Dalirrooyfard
 EECS & CSAIL
 MIT
 Cambridge, MA, USA
 minad@mit.edu

Ce Jin
 EECS & CSAIL
 MIT
 Cambridge, MA, USA
 cejin@mit.edu

Virginia Vassilevska Williams
 EECS & CSAIL
 MIT
 Cambridge, MA, USA
 virgi@mit.edu

Nicole Wein
 DIMACS
 Rutgers University
 New Brunswick, NJ, USA
 nicole.wein@rutgers.edu

Abstract—We study the approximability of two related problems on graphs with n nodes and m edges: *n -Pairs Shortest Paths* (n -PSP), where the goal is to find a shortest path between $O(n)$ prespecified pairs, and *All Node Shortest Cycles* (ANSC), where the goal is to find the shortest cycle passing through each node. Approximate n -PSP has been previously studied, mostly in the context of *distance oracles*. We ask the question of whether approximate n -PSP can be solved faster than by using distance oracles or All Pair Shortest Paths (APSP). ANSC has also been studied previously, but only in terms of exact algorithms, rather than approximation.

We provide a thorough study of the approximability of n -PSP and ANSC, providing a wide array of algorithms and conditional lower bounds that trade off between running time and approximation ratio.

A highlight of our conditional lower bounds results is that for any integer $k \geq 1$, under the combinatorial $4k$ -clique hypothesis, there is no combinatorial algorithm for unweighted undirected n -PSP with approximation ratio better than $1 + 1/k$ that runs in $O(m^{2-2/(k+1)}n^{1/(k+1)-\varepsilon})$ time. This nearly matches an upper bound implied by the result of Agarwal (2014).

Our algorithms use a surprisingly wide range of techniques, including techniques from the girth problem, distance oracles, approximate APSP, spanners, fault-tolerant spanners, and link-cut trees.

A highlight of our algorithmic results is that one can solve both n -PSP and ANSC in $\tilde{O}(m + n^{3/2+\varepsilon})$ time¹ with approximation factor $2 + \varepsilon$ (and additive error that is function of ε), for any constant $\varepsilon > 0$. For n -PSP, our conditional lower bounds imply that this approximation ratio is nearly optimal for any subquadratic-time combinatorial algorithm. We further extend these algorithms for n -PSP and ANSC to obtain a time/accuracy trade-off that includes near-linear time algorithms.

Additionally, for ANSC, for all integers $k \geq 1$, we extend the very recent almost k -approximation algorithm for the girth problem that works in $\tilde{O}(n^{1+1/k})$ time [Kadria et al. SODA'22], and obtain an almost k -approximation algorithm for ANSC in $\tilde{O}(mn^{1/k})$ time.

Index Terms—graph algorithms, fine-grained complexity, approximation algorithms

I. INTRODUCTION

The focus of this paper is two basic problems concerning distances in graphs: the *n -Pairs Shortest Paths* problem and

The full version [1] of this paper is available at <https://arxiv.org/abs/2204.03076>.

¹ \tilde{O} hides sub-polynomial factors.

the *All-Nodes Shortest Cycles* problem.

n -Pairs Shortest Paths (n -PSP). Given a (weighted or unweighted, directed or undirected) graph with n nodes and m edges, and a set of pairs of vertices (s_i, t_i) for $1 \leq i \leq O(n)$, compute the distance from s_i to t_i for every i . (For ease of notation, we denote this problem n -PSP even though the number of pairs is not exactly n , rather it is $O(n)$.)

All-Nodes Shortest Cycles (ANSC). Given a (weighted or unweighted, directed or undirected) graph with n nodes and m edges, compute for each vertex v , the length of the shortest cycle containing v , denoted $SC(v)$.

As we will show, these two problems are very similar in some ways and fundamentally different in other ways. We first provide some background for the n -PSP problem.

a) The n -PSP problem.: The n -PSP problem was first explicitly studied in the 90s. Aingworth, Chekuri, Indyk and Motwani [2] obtained an additive 2-approximation in time $\tilde{O}(n^2)$. The other early work on this problem has been subsequently subsumed by later results for distance oracles [3], [4].

As far as we are aware, the n -PSP problem has not been explicitly studied since the 90s. However, other distance-related problems have been studied in the setting where one only cares about the distances between prespecified vertex pairs, such as pairwise distance preservers, pairwise spanners, which were first studied by Coppersmith and Elkin [5] and extensively studied thereafter, as well as pairwise reachability preservers [6].

Now, we will provide some motivation for studying the n -PSP problem. Perhaps the most classical distance problem is All-Pairs Shortest Paths (APSP). APSP can be solved in directed graphs with non-negative edge weights in time $\tilde{O}(mn)$ simply by running Dijkstra's algorithm from each vertex. For undirected unweighted graphs, APSP can be solved using matrix multiplication in time $\tilde{O}(n^\omega)$ [7], where $2 \leq \omega < 2.373$ is the matrix multiplication exponent [8]. For directed unweighted graphs, APSP can be solved in time $\tilde{O}(n^{2.529})$ [9] (the bound can be slightly improved by plugging in a better rectangular matrix multiplication [10]). For very large graphs,

these running times can be prohibitive; even just writing down the output of size n^2 can be too slow.

For many applications in both theory and practice, computing *all* of the distances in the graph is overkill, and instead we only care about *some* of the distances (e.g. multi-source multi-sink routing [11], many-to-many shortest paths [12], etc.). We ask a question that has been asked many times before:

Can we compute some distances in a graph faster than computing all distances?

This question has been approached from various angles:

- The simplest approach to this question is perhaps to compute all distances from a single source. The Single-Source Shortest Paths (SSSP) problem can indeed be solved much faster than APSP (by Dijkstra's algorithm in $O(m + n \log n)$ time), but has the obvious drawback that all of the distances computed have the same source.
- Another approach towards this question, is to compute only the *extremal* distances in the graph, that is, the diameter, radius, and eccentricities (the largest distance from each vertex in the graph). For these problems, there are conditional lower bounds that rule out subquadratic time *exact* algorithms for sparse graphs [13], but there has been extensive work on *approximating* these parameters quickly (see e.g. [14]). This approach has the drawback that it only concerns extremal distances, and one might wish to compute or approximate an *arbitrary* set of distances.
- Another approach towards this question is to construct a *distance oracle*, a data structure with subquadratic space that allows one to quickly query (approximate) distances. Distance oracles are designed for the setting where we wish to know some arbitrary set of distances, but we do not know *a priori* which distances. Distance oracles have been extensively studied, and various trade-offs between approximation ratio and running time are known (See Section I-A for more detailed discussion of distance oracles).

In contrast to distance oracles, we ask the question: what if we *do* know *a priori* which distances we wish to compute? The *n*-PSP problem is precisely this problem, where we have $O(n)$ prespecified vertex pairs. We ask the question of whether we can achieve algorithms for *n*-PSP that are faster than the algorithms directly implied by known distance oracles.

In this work, we will show that this question has different answers in different regimes. For example,

- In the regime of $(1 + 1/k)$ -approximations, we show that the *n*-PSP algorithm directly implied by Agarwal's distance oracle [15] has nearly optimal running time, under the combinatorial $4k$ -clique hypothesis.
- For $(2 + \varepsilon, \beta)$ -approximation², we show an *n*-PSP algorithm that runs faster than directly applying the state-of-the-art distance oracle of Chechik and Zhang [16].

²An (α, β) approximation algorithm means that the algorithm has multiplicative error α and additive error β .

Now, we turn our attention to ANSC.

b) The ANSC problem.: The ANSC problem was first studied by Yuster [17], who gave a randomized algorithm for undirected graphs with integer weights from 1 to M , in time $\tilde{O}(\sqrt{M}n^{(\omega+3)/2})$. Later, Sankowski and Węgrzycki [18], and independently Agarwal and Ramachandran [19], showed that for unweighted *directed* graphs there is a deterministic $\tilde{O}(n^\omega)$ time algorithm. Agarwal and Ramachandran [19] also gave a reduction from the *Replacement Paths* problem to weighted directed ANSC. In the Replacement Paths problem, we are given a graph and a shortest path P between two vertices s and t , and the goal is to find for every edge $e \in P$, a shortest path from s to t that avoids e . The reduction of [19] increases the edge weights by a factor of n , and was subsequently improved to preserve the range of edge weights by Chechik and Nechushtan [20].

Despite this prior work on the exact version of ANSC, as far as we know, we are the first to study the approximability of ANSC.

In this work, we show various algorithmic results for the ANSC problem. For example,

- We show an almost k -approximation algorithm for ANSC with running time comparable to the best known k -approximation girth algorithm.
- We show a $(2 + \varepsilon, \beta)$ -approximation algorithm for ANSC with subquadratic running time.

A. Results for *n*-PSP and ANSC implied by prior work

We begin with two observations that relate *n*-PSP and ANSC. The proofs of these observations are in the appendix. The first observation is a reduction from exact *n*-PSP to exact ANSC in weighted graphs. The second observation is a reduction from ANSC to *n*-PSP in directed graphs that works for the approximation setting with any finite approximation factor.

A $T(n, m)$ -time algorithm solving weighted undirected ANSC exactly implies a $T(n, m)$ -time algorithm for solving weighted undirected *n*-PSP exactly.

A $T(n, m)$ -time algorithm solving (unweighted) directed *n*-PSP with any finite approximation factor $\alpha \geq 1$ implies a $T(n, m)$ -time algorithm for solving (unweighted) directed ANSC with approximation factor α .

One simple way to obtain approximation algorithms for directed or undirected *n*-PSP and directed ANSC is to use known algorithms for approximate All-Pairs Shortest Paths (APSP). The running times of these algorithms, however, will always be at least $\Omega(n^2)$ due to the size of the output of APSP. If there is an algorithm for approximate APSP in $T(n, m)$ time for directed or undirected graphs with n nodes and m edges, then we can approximate directed or undirected (respectively) *n*-PSP in $O(n) + T(n, m)$ time with the same accuracy by looking up the $O(n)$ input pairs in the output of the APSP algorithm. We can approximate directed ANSC in $O(n^2) + T(n, m)$ time by computing $\min_u \hat{d}(v, u) + \hat{d}(u, v)$ for every v , where $\hat{d}(\cdot, \cdot)$ is the distance estimate that the APSP algorithm outputs. The approximation guarantee for

ANSC will have the same multiplicative error as APSP, and the additive error will double. Since the output of APSP is of size $\Theta(n^2)$, the second term in these running times is dominant and so we can approximate both directed or undirected n -PSP and directed ANSC in $O(T(n, m))$ time. We note that it is not clear how to get an algorithm for undirected ANSC directly from APSP, as Observation I-A works only for directed graphs. Next, we outline the known algorithms for APSP.

We have already outlined the known algorithms for exact APSP. As for approximation algorithms for APSP, in directed or undirected graphs with non-negative edge weights, for any $\varepsilon > 0$, Zwick [9] gave a $(1 + \varepsilon)$ approximation time algorithm in time $O(\frac{n^\omega}{\varepsilon} \log(W))$ where W is the largest edge weight. In the undirected setting, APSP in graphs with integer weights in $[-W, W]$ can be solved in $O(n^\omega \log(W))$ [21]. For any $\varepsilon > 0$, [22] gives a $(1 + \varepsilon)$ approximation algorithm in $O(\frac{n^\omega}{\varepsilon} \text{polylog}(\frac{n}{\varepsilon}))$ time, improving upon Zwick's algorithm for large weights. There are many more algorithms for approximating APSP [2], [23]–[26], however since the focus of our paper is on subquadratic-time algorithms we do not describe them in detail.

Another simple way to obtain approximation algorithms for undirected n -PSP, is to use an approximate distance oracle (DO). A DO is a data structure that allows one to query distances. The parameters of interest in a DO are preprocessing time, query time, space, and (multiplicative and additive) approximation ratio. Given a DO with preprocessing time $p(m, n)$, query time $q(m, n)$, one can obtain an algorithm for n -PSP with the same approximation ratio, in time $p(m, n) + O(n) \cdot q(m, n)$ simply by querying all of the input pairs. Unlike algorithms for n -PSP that are based on APSP, algorithms based on DOs do not have an inherent running time of $\Omega(n^2)$.

We focus on DOs with subquadratic preprocessing time. For any integer $k \geq 2$, the Thorup-Zwick DO [27] has preprocessing time $O(kmn^{1/k})$, size $O(n^{1+1/k})$ with query time $O(k)$ and approximation factor $(2k - 1)$. We can use this DO to obtain a $(2k - 1)$ -approximation algorithm for weighted undirected n -PSP in time $O(kmn^{1/k})$. Patrascu et al [28] also extends the Thorup-Zwick DO to fractional values of k . Additionally, [29] gives a DO with preprocessing time $O(mn^{2/3})$ and constant query time that returns a path of length at most $2d + 1$ when queried for a pair at distance d . This gives us a $(2, 1)$ -approximation algorithm in $O(mn^{2/3})$ time for n -PSP. On the other side of the time/accuracy trade-off, Agarwal [15] gives a DO that yields for any integer $k \geq 1$, a $(1 + \frac{1}{k})$ -approximation algorithm for n -PSP in time $\tilde{O}(m^{2-2/(k+1)}n^{1/(k+1)})$, as well as a $(1 + \frac{1}{k+0.5})$ -approximation algorithm for n -PSP in time $\tilde{O}(m^{2-3/(k+2)}n^{2/(k+2)})$ (see full version [1] for explanation). Additionally, there are DOs with processing times that have additive dependence between n and m . For any integer $k \geq 1$, Wulff-Nilsen [23] gives a $(2k - 1)$ approximate distance oracle with preprocessing time $O(\sqrt{km} + n^{1+c/\sqrt{k}})$ for a constant $c = 9 + 3\sqrt{13}$, and query time $O(k)$. This gives a $(2k - 1)$ approximation algorithm for n -PSP in $O(\sqrt{km} + n^{1+c/\sqrt{k}})$

time. Very recently, Chechik and Zhang [16] obtained a constant query time $(2 + \varepsilon, \beta)$ -approximate distance oracle with subquadratic preprocessing time $\tilde{O}(m + n^{5/3+\varepsilon})$, which immediately implies a $(2+\varepsilon, \beta)$ -approximate n -PSP algorithm in $\tilde{O}(m + n^{5/3+\varepsilon})$ time.

Note that we cannot use Observation I-A to obtain an algorithm for ANSC since it only works in the exact setting, and it is not clear how to use distance oracles in general to solve ANSC in the undirected setting. Moreover, there are no non-trivial distance oracles for the directed setting [27].

Thus, past work doesn't give any subquadratic approximation algorithms for ANSC in directed or undirected graphs. For undirected n -PSP we get the above results from distance oracles, but it is not clear if this is the best one can do.

From the lower bounds side, ANSC is closely related to Girth (the problem of finding the smallest cycle in the graph), and any conditional lower bounds for Girth immediately carry over to ANSC. A previously known lower bound for the Girth in directed graphs states that under the k -Cycle Hypothesis, any better than 2-approximation for Girth requires time $m^{2-o(1)}$ [30].

B. Our Results

We provide a thorough study of the approximability of n -PSP and ANSC, providing a wide array of algorithms and conditional lower bounds that trade off between running time and approximation.

Before stating our results we provide the main hardness assumptions that we use for our conditional lower bounds.

1) *Main Hardness Assumptions*: We stress that our conditional lower bounds are based on well-established hardness assumptions in fine-grained complexity. We obtain hardness results based on a number of different assumptions, but for the sake of clarity, we only list the two central ones here. For a complete list of the assumptions we use and the associated hardness results see full version [1].

Our first hypothesis concerns *combinatorial algorithms* for k -clique detection, and has been used as a hardness hypothesis in [31]–[36]. By “combinatorial” we mean algorithms that do not use the heavy machinery of Fast Matrix Multiplication.

Hypothesis 1 (Combinatorial k -Clique Hypothesis). *Let $k \geq 3$ be a constant integer. In the word-RAM model with $O(\log n)$ bit words, there is no $O(n^{k-\varepsilon})$ time combinatorial algorithm for k -clique detection, for any constant $\varepsilon > 0$.*

The Combinatorial 3-Clique Hypothesis is also called *Combinatorial Dense Triangle Hypothesis*, which is equivalent to the *Combinatorial Boolean Matrix Multiplication Hypothesis* (see full version [1]).

Our next hypothesis is used for our main conditional lower bound result. This hypothesis was introduced in [35] and concerns (not necessarily combinatorial) algorithms for k -clique detection in hypergraphs. It has been widely used as a hardness assumption [37]–[46].

Hypothesis 2 ((k, r)-Hyperclique Hypothesis). *Let $k > r \geq 3$ be a constant integer. In the word-RAM model with $O(\log n)$*

bit words, there is no $O(n^{k-\varepsilon})$ time algorithm for k -clique detection in r -uniform hypergraphs, for any constant $\varepsilon > 0$.

2) *Hardness Results:* Our hardness results for n -PSP are shown in Table I, and those for ANSC are shown in Table II. For the sake of clarity we only describe our most central hardness results in words here. For a detailed statement of all of our hardness results see full version [1].

Most of our hardness results are actually stronger than needed for Pairwise Distance and ANSC. In particular, they also apply for weaker problems, specifically the *extremal* versions of n -PSP and ANSC where the output is either the minimum or maximum among the $O(n)$ values outputted. In particular, the *girth* is the length of the smallest cycle in the graph, that is, the minimum value in the output of ANSC. We define the *cycle diameter* as the largest value in the output of ANSC, that is, the maximum over all vertices v of the length of the smallest cycle through v . Analogously for n -PSP, given a graph and $O(n)$ pairs of vertices (s_i, t_i) , we define the *n -pairs minimum distance* as $\min_i d(s_i, t_i)$, and the *n -pairs diameter* as $\max_i d(s_i, t_i)$.

We begin with a simple hardness result for n -Pairs Minimum Distance. Using a simple reduction, we get the following theorem is against a $(2-\varepsilon)$ -approximation by a combinatorial algorithm.

Theorem I.1. *Under the Combinatorial Dense Triangle Hypothesis, any better than 2-approximation combinatorial algorithm for n -Pairs Minimum Distance requires $n^{3/2-o(1)}$ time.*

Theorem I.1 has two main drawbacks: the running time is not as high as we would like, and it is only for combinatorial algorithms. We overcome both of these drawbacks. We achieve stronger running time bounds under a generalized version of the Combinatorial Dense Triangle Hypothesis: the Combinatorial k -Clique Hypothesis. We also remove the “combinatorial” condition of Theorem I.1 under the (k, r) -Hyperclique Hypothesis. To achieve both of these goals, as well as establish a wide range of time/accuracy trade-off lower bounds for both combinatorial and non-combinatorial algorithms, some of which are nearly tight, we introduce the following general theorem. This theorem is our most technically substantial conditional lower bound. After stating the theorem, we will highlight some of its corollaries.

Theorem I.2. *For integers r, k, t satisfying $k-1 \geq t+1 \geq r \geq 2$, let*

$$D = 2r(t+1) - (2r-3)k,$$

and suppose $k < D$. Then the following holds:

Given a k -(hyper)-clique instance on an n -vertex r -uniform (hyper)-graph G , we can reduce it (in linear time) to an unweighted undirected n -Pairs Minimum Distance instance with $O(kn^t)$ vertices and $O(kn^{t+1})$ edges, such that:

- 1) *If G contains a k -(hyper)-clique, then the n -pairs minimum distance equals k .*
- 2) *If G does not contain a k -(hyper)-clique, then the n -pairs minimum distance is at least D .*

Our first two corollaries of Theorem I.2 concern the graph (not hypergraph) version of Theorem I.2, and are under the Combinatorial k -Clique Hypothesis.

Corollary I.1 establishes a time/accuracy trade-off against algorithms with faster running times and higher approximation ratio, while Corollary I.2 establishes a time/accuracy trade-off against algorithms with better approximation ratios and slower running times.

Corollary I.1. *For $k \geq 4$, assuming the Combinatorial k -Clique Hypothesis, there is no combinatorial algorithm for unweighted undirected n -Pairs Minimum Distance with approximation ratio better than $(3-4/k)$ in $m \cdot n^{1/(k-2)-\varepsilon}$ time or $m + n^{k/(k-2)-\varepsilon}$ time, for any constant $\varepsilon > 0$.*

Corollary I.2. *For $k \geq 1$, assuming the Combinatorial $4k$ -Clique Hypothesis, there is no combinatorial algorithm for unweighted undirected n -Pairs Minimum Distance with approximation ratio better than $(1+1/k)$ in $n^{2-\varepsilon}$ time or $m \cdot n^{1-1/(2k)-\varepsilon}$ time or $m^{2-2/(k+1)} \cdot n^{1/(k+1)-\varepsilon}$ time, for any constant $\varepsilon > 0$.*

Corollary I.3. *For $k \geq 1$, assuming the Combinatorial $(4k+2)$ -Clique Hypothesis, there is no combinatorial algorithm for unweighted undirected n -Pairs Minimum Distance with approximation ratio better than $(1+1/(k+0.5))$ in $n^{2-\varepsilon}$ time or $m \cdot n^{1-1/(2k+1)-\varepsilon}$ time or $m^{2-3/(k+2)} \cdot n^{2/(k+2)-\varepsilon}$ time, for any constant $\varepsilon > 0$.*

Corollary I.2 is nearly tight with previously known algorithms for n -PSP in the following sense. For any integer $k \geq 1$, there is a $(1+1/k)$ -approximation for n -PSP in time $\tilde{O}(m^{2-2/(k+1)} n^{1/(k+1)})$ [15], while Corollary I.2 says that there is no better than $(1+1/k)$ -approximation in $\tilde{O}(m^{2-2/(k+1)} n^{1/(k+1)-\varepsilon})$. That is, one conditionally cannot simultaneously improve both the running time and the approximation factor of the known algorithms, for any k .

Similarly, Corollary I.3 is also nearly tight with another algorithm for n -PSP implied by [15]’s results, which has approximation ratio $1+1/(k+0.5)$ and running time $\tilde{O}(m^{2-3/(k+2)} n^{2/(k+2)})$. Corollary I.3 says that one conditionally cannot simultaneously improve both the running time and the approximation factor of this algorithm for any k .

Our final corollary concerns the hypergraph version of Theorem I.2 and is under the (k, r) -Hyperclique Hypothesis (for $r \geq 3$). Unlike, the above two corollaries, the following corollary is for not necessarily combinatorial algorithms.

Corollary I.4. *For $k \geq 4$, assuming the $(k, 3)$ -Hyperclique Hypothesis, there is no algorithm for unweighted undirected n -Pairs Minimum Distance with approximation ratio better than $(3-6/k)$ in $n^{k/(k-2)-\varepsilon}$ or $mn^{1+1/(k-2)-\varepsilon}$ time, for any constant $\varepsilon > 0$.*

Assuming the $(4, 3)$ -Hyperclique Hypothesis, Corollary I.4 rules out algorithms with approximation ratio better than $3/2$ in $n^{2-\varepsilon}$ time or $mn^{1/2-\varepsilon}$ time, for any constant $\varepsilon > 0$. This is the choice of parameters with the best possible running time.

Approximation	Running time LB	Theorem	Hypothesis	Comments
$5/3 - \varepsilon$	$m^{1+\delta-o(1)}$	[1]	Sparse Triangle	
$5/3 - \varepsilon$	$n^{\omega-o(1)}$	[1]	Dense Triangle	
$5/3 - \varepsilon$	$n^{\omega-o(1)}$	[1]	Simplicial Vertex	for n -Pairs Diameter
$2 - \varepsilon$	$n^{3/2-o(1)}$	Thm I.1	Comb. BMM	
$2 - \varepsilon$	$m + n^{2-o(1)}$	Cor I.1	Comb. 4-clique	nearly matches Thm I.3
$3 - 4/k - \varepsilon$	$m + n^{k/(k-2)-o(1)}$	Cor I.1	Comb. k -clique	
$1 + 1/k - \varepsilon$	$m^{2-2/(k+1)} n^{1/(k+1)-o(1)}$	Cor I.2	Comb. 4 k -clique	nearly matches [15]
$1 + 1/(k+0.5) - \varepsilon$	$m^{2-3/(k+2)} n^{2/(k+2)-o(1)}$	Cor I.3	Comb. $(4k+2)$ -clique	nearly matches [15]
$3/2 - \varepsilon$	$mn^{1/2-o(1)}$	Cor I.4	$(4, 3)$ -hyperclique	
$3 - 6/k - \varepsilon$	$m + n^{k/(k-2)-o(1)}$	Cor I.4	$(k, 3)$ -hyperclique	
any finite	$m^{2-o(1)}$	[1]	k -cycle	directed graphs

TABLE I

Conditional Lower Bounds for (unweighted) n -PSP. All results are for undirected graphs unless otherwise specified. All results work for n -Pairs Minimum Distance unless otherwise specified. See the body of the text for details about the near-tightness of some of our conditional lower bounds.

Approximation	Running time LB	Theorem	Hypothesis	Comments
$4/3 - \varepsilon$	$m^{1+\delta-o(1)}$	[1]	Sparse Triangle	for Girth
$4/3 - \varepsilon$	$n^{\omega-o(1)}$	[1]	Dense Triangle	for Girth
$7/5 - \varepsilon$	$n^{\omega-o(1)}$	[1]	Simplicial Vertex	for Cycle Diameter
$3/2 - \varepsilon$	$m^{4/3-o(1)}$	[1]	All-Edges Sparse Triangle	
$3/2 - \varepsilon$	n^2	[1]	unconditional	

TABLE II
Conditional Lower Bounds for (unweighted) undirected ANSC.

As we discuss later, this is nearly tight with our algorithm from Theorem I.3.

By setting k to be large in Corollary I.4, we obtain lower bounds with approximation ratio larger than 2 and close to 3. We note that this is the first known lower bound with approximation ratio higher than 2 for *any distance problem* except for the *ST-Diameter* problem, but unlike in n -PSP, the number of vertex pairs one considers in *ST-Diameter* is much larger than the running time lower bounds.

We also note that by setting $k = 6$ in Corollary I.4, we get the same bound as Theorem I.1, but for not necessarily combinatorial algorithms.

Comparison with independent work [47].: Very recently, Abboud, Bringmann, Khoury, and Zamir [47] proposed the “cycle-removal” framework, and used it to obtain new conditional lower bounds for various graph problems related to approximating distances or girth. In particular, they showed super-linear lower bounds on the preprocessing time of k -approximate distance oracles. Their lower bounds also applied to the offline setting of distance oracle queries, which is almost the same as the n -PSP problem we considered here, except that they did not fix the number of query pairs to be n . Their results imply that, under either 3-SUM hypothesis or APSP hypothesis, for any constant $k \geq 4$, n -PSP does not have k -approximation algorithms in $m^{1+c/k}$ time, where $c > 0$ is some universal constant. Their result has the right form $m^{1+\Theta(1/k)}$, but it appears difficult to obtain the best possible constant c on the exponent using their framework.

The main difference between their lower bound results and ours (Corollaries I.1 to I.3) is that we focus on approximation ratio much closer to 1, such as $1 + 1/k - \varepsilon$, while they focus on arbitrarily large constant approximation ratio. Corollary I.2

and Corollary I.3 nearly match the known upper bounds [15], without losing constant factors on the exponent. The downside of Corollaries I.1 to I.3 is that they only work against combinatorial algorithms.

Some of our other corollaries (such as Corollary I.4) obtained $m + n^{1+1/(k-2)-o(1)}$ lower bounds n -PSP with approximation ratio $3 - \Theta(1/k)$. They are subsumed by [47] when the constant k is large enough.

3) Algorithmic Results: We investigate approximation algorithms for both the n -PSP and ANSC problems in both directed and undirected settings in n -node m -edge graphs. Additionally, we are interested in the dependency between m and n in the running time of our algorithms. We first present algorithms where the running time shows a *multiplicative dependency* between n and m . Then we investigate approximation algorithms for n -PSP and ANSC whose running time has *additive* dependence between n and m , in particular running times of the form $m + n^{2-\varepsilon}$. Algorithms of this form are desirable in part because they yield near-linear time algorithms for dense enough graphs. Moreover, algorithms of the form $m + n^{2-\varepsilon}$ have been studied for a variety of problems, for instance in distance oracles [16], [23], and recent results on bipartite matching and related problems [48]. Another motivation for studying such algorithms is that known undirected Girth algorithms do not have any multiplicative dependency on m , and so we ask how crucial this multiplicative dependency is for undirected ANSC and n -PSP. (Known *directed* Girth algorithms, however do have multiplicative dependency on m .)

We let an (α, β) -approximation denote an approximation algorithm that outputs an estimate \hat{x} for x such that $x \leq \hat{x} \leq \alpha \cdot x + \beta$.

a) *The n -PSP problem.*: Our algorithmic results for n -PSP are shown in Table III. Note that there is no constant factor approximation algorithm for the directed case (see full version [1]), and hence all of our algorithmic results for n -PSP are for the undirected case.

First we give a straightforward nearly $2k - 2$ -approximation for n -PSP, and we leave it as an open problem whether one can achieve a k -approximation with similar running time.

Particularly related to our work, Chechik and Zhang [16] obtained various distance oracles for unweighted undirected graphs with subquadratic construction time and constant query time. Their result immediately implies a $(2 + \varepsilon, \beta)$ -approximate n -PSP algorithm in $\tilde{O}(m + n^{5/3+\varepsilon})$ time. In this work we obtain a faster algorithm for n -PSP, stated in the following theorem.

Theorem I.3. *Given an n -node m -edge undirected unweighted graph G and vertex pairs (s_i, t_i) for $1 \leq i \leq O(n)$, for any constant $\varepsilon > 0$, there is a randomized algorithm that computes a $(2 + \varepsilon, \beta)$ -approximation for n -PSP in $\tilde{O}(n^{3/2+\varepsilon} + m)$ time with high probability, for some constant β depending on ε .*

Theorem I.3 is nearly tight with our conditional lower bound from Corollary I.1 in the sense that Corollary I.1 conditionally rules out a $(2 - \varepsilon)$ -approximation in time $n^{2-\Omega(1)}$, while Theorem I.3 provides a $(2 + \varepsilon, f(\varepsilon))$ -approximation in time polynomially faster than $O(n^2 + m)$.

b) *The ANSC problem.*: Our algorithmic results for ANSC are shown in Table IV.

For directed graphs, we provide approximation algorithms for ANSC, showing a strong separation between ANSC and n -PSP in the directed case. To obtain our algorithms for directed ANSC, we generalize previously known results from Girth by Dalirrooyfard and Vassilevska W. [30], with a slight loss in the accuracy and the running time. These results are stated in Table IV and the appendix.

We now move to the case of undirected graphs. Recall that from Table III, we give a nearly $2k - 2$ -approximation for n -PSP. For ANSC, however, we are able to achieve the better approximation ratio of $(k + \varepsilon)$, as stated in the following theorem.

Theorem I.4. *Given an n -node m -edge undirected graph G with edge weights in $\{1, \dots, M\}$, a constant $\varepsilon > 0$ and an integer $k \geq 3$, there is a randomized algorithm that computes a $(k + \varepsilon)$ -approximation for ANSC in $\tilde{O}(mn^{1/k} \log(M))$ time with high probability.*

Very recently, a result similar to Theorem I.4 was shown for the Girth problem [49]: an almost k -approximation algorithm for the undirected girth in $\tilde{O}(n^{1+1/k})$ time. Our running time for ANSC is instead $\tilde{O}(mn^{1/k})$ and this dependence on m is not unexpected since for Girth, one generally only runs Dijkstra's algorithm until finding a cycle which takes $\tilde{O}(n)$ time, whereas for ANSC, we execute Dijkstra's algorithm to completion.

Now, we move to algorithms for ANSC of the form $m + n^{2-\varepsilon}$. We consider these to be our main algorithmic results.

We begin with a “proof of concept” algorithm which shows that there is indeed an algorithm for ANSC with constant multiplicative and additive factors in time $m + n^{2-\varepsilon}$ for constant ε .

Theorem I.5. *Given an n -node m -edge undirected unweighted graph G , there is a randomized algorithm that computes a $(6, 1)$ -approximation for ANSC in $\tilde{O}(m + n^{2-1/6})$ time with high probability.*

We will significantly improve upon Theorem I.5 in running time and multiplicative factor in our next result. However, our next algorithm does not *strictly* improve upon Theorem I.5 partially due to its additive error of only 1.

Our goal is to reduce the multiplicative approximation ratio as much as possible, with the goal of getting it down to nearly 2, to match our above algorithm for n -PSP.

Theorem I.6. *Given an n -node m -edge undirected unweighted graph G and a constant $\varepsilon > 0$, there is a randomized algorithm that computes a $(2 + \varepsilon, \beta)$ -approximation for ANSC in $\tilde{O}(n^{1.5+\varepsilon} + m)$ time, where β is a constant depending only on ε .*

We also use fast matrix multiplication to obtain improvement results for the ST -shortest paths problem, which is a special case of n -PSP. They are included in the full version [1].

II. TECHNICAL OVERVIEW

We use many different techniques that were originally designed for a range of different problems and data structures, such as girth, APSP, distance oracles, spanners, fault-tolerant spanners, the simplicial vertex problem, and link-cut trees. The applicability of some of these problems to approximate n -PSP and ANSC is perhaps unexpected. For example, it is not clear how something like a fault-tolerant spanner would be useful in a setting that does not involve faulty vertices or edges.

Although we pull together results from a variety of different problems, our results are not “just” an application of prior techniques. In the following overview of our techniques, we provide an overview of many of our results, choosing to highlight certain results that require significantly new ideas from prior work. In particular, we highlight a collection of lower bounds for n -PSP, as well as our collection of approximation algorithms for ANSC with running times of the form $\tilde{O}(m + n^{2-\varepsilon})$.

A. Conditional Lower Bounds

Our conditional lower bounds are from standard hardness assumptions for basic problems such as triangle detection, k -cycle, and k -clique. Many of our conditional lower bounds are quite straightforward reductions from these problems. One of our conditional lower bounds, however, is more technically substantial, and we highlight it next.

Approximation	Running time	Theorem	Comments
$(2 + \varepsilon, f(\varepsilon))$	$m + n^{3/2+\varepsilon}$	Thm I.3	$\varepsilon > 0$, some function f , nearly matches Cor I.1
$2k - 2$	$mn^{1/k}$	[1]	integer $k \geq 2$
$(2k - 1) \cdot (2k - 2)$	$m + n^{1+2/k}$	[1]	integer $k \geq 2$
2	$m \cdot n^{(1+\omega)/8}$	[1]	for ST -Shortest Paths, $ S , T = O(\sqrt{n})$

TABLE III

A summary of all of our approximation algorithms for n -PSP in undirected unweighted graphs. (Some results also work for weighted graphs as stated in each theorems.) All running times above are within $\text{polylog}(n)$ factors.

Approximation	Running time	Theorem	Comments
$2k + 1 + \varepsilon$	mn^{α_k}	[1]	directed graphs, α_k solves $\alpha_k(1 + \alpha_k)^{k-1} = 1 - \alpha_k$
$2 + \varepsilon$	$mn^{1/2}$	[1]	directed graphs, nearly matches Thm 5.1 of [30]
$k + \varepsilon$	$mn^{1/k}$	Thm I.4	$\varepsilon > 0$ and integer $k \geq 2$
nearly $1 + 1/(k-1)$	$m^{2-2/k}n^{1/k}$	[1]	integer $k \geq 2$
(6, 1)	$m + n^{2-1/6}$	Thm I.5	
$(2 + \varepsilon, \beta)$	$m + n^{1.5+\varepsilon}$	Thm I.6	$\varepsilon > 0$ and β is a function of ε
$(k^2, k^3 2^{k+1})$	$m + n^{1+2/k}$	[1]	integer $k \geq 2$

TABLE IV

A summary of all of our approximation algorithms for unweighted ANSC. (Some results also work for weighted graphs as stated in each theorem.) All running times above are within $\text{polylog}(n)$ factors. All results are for undirected graphs unless otherwise specified.

1) *Highlight: Hardness of n -PSP from k -(Hyper)Clique Hypotheses:* For the undirected unweighted n -PSP problem, by adapting known techniques one could only prove fine-grained hardness for approximation ratio less than 2. To overcome this issue, we give an interesting and novel reduction from the Combinatorial k -Clique Hypothesis to the n -Pairs Minimum Distance problem, which not only yields lower bounds for approximation ratio higher than 2, but also gives tight bounds that match some of our algorithms in the low-approximation regime. We believe this powerful reduction will inspire more fine-grained hardness results for related problems.

As an illustrative example, we describe the case of $k = 4$. Suppose we are given a 4-clique instance on a 4-partite graph G with vertex partition V_1, V_2, V_3, V_4 where $|V_i| = n$. In our reduction we create a 5-partite graph G' with vertex partition $V_{12}, V_{23}, V_{34}, V_{41}, V'_{12}$ from left to right, where each part contains n^2 vertices, and we will connect edges only between adjacent parts. The vertices in V_{12} (and V'_{12}) are indexed by vertex pairs $(v_1, v_2) \in V_1 \times V_2$ from the input graph G , and V_{23}, V_{34}, V_{41} are similarly (according to the subscripts) indexed by vertex pairs from G . For every $v_1 \in V_1, v_2 \in V_2, v_3 \in V_3$, we connect an edge between the two vertices $(v_1, v_2) \in V_{12}, (v_2, v_3) \in V_{23}$ in G' , if and only if (v_1, v_2, v_3) form a 3-clique in G . We similarly add edges between V_{23}, V_{34} , between V_{34}, V_{41} , and between V_{41}, V'_{12} . The resulting graph G' has $N = 5n^2$ vertices and at most $M = 4n^3$ edges.

If G contains a 4-clique (v_1, v_2, v_3, v_4) , then in G' there exists a length-4 path from $(v_1, v_2) \in V_{12}$ to $(v_1, v_2) \in V'_{12}$, by simply going from left to right along the edges specified by v_1, v_2, v_3, v_4 . The interesting part is the converse case: when G does not contain a 4-clique, what is the shortest possible length of any path from $(v_1, v_2) \in V_{12}$ to $(v_1, v_2) \in V'_{12}$? Note that this question amounts to solving an n -Pairs Minimum Distance instance on G' . The answer cannot be 4, since a length-4 path would immediately recover a 4-clique (containing (v_1, v_2) in

G). The next smallest length is 6 (with one backward step, and one extra forward step), and it turns out that a length-6 path would also have to recover a 4-clique in G : for example, the following length-6 path $(v_1, v_2) \rightarrow (v_2, v_3) \rightarrow (v_3, v_4) \rightarrow (v'_2, v_3) \rightarrow (v_3, v'_4) \rightarrow (v'_4, v_1) \rightarrow (v_1, v_2)$ implies the existence of the 4-clique (v_1, v_2, v_3, v'_4) ; the other possibilities of length-6 paths can be similarly verified. Hence, the smallest possible answer is 8. Then, any better than 2-approximation combinatorial algorithm for n -Pairs Minimum Distance with $m \cdot n^{1/2-\varepsilon}$ running time would be able to distinguish these two cases of distance 4 or ≥ 8 and hence solve the 4-clique instance, in $M \cdot N^{1/2-\varepsilon} \leq O(n^{4-\varepsilon/2})$ time, contradicting the combinatorial 4-clique hypothesis. This lower bound nearly matches the upper bound from [15].

The reduction described above can be generalized to larger k , but naive ways to prove the lower bound on the shortest distance would require an exhaustive case analysis, which does not work for general values of k . To overcome this issue, we provide a clean combinatorial argument that can pinpoint the vertices participating in a k -clique when there is a too short path. This combinatorial argument is highly extendable: It yields lower bounds with approximation ratio that increases with k , and it even allows us to capture *hypercliques* rather than cliques. As a result, we can prove *non-combinatorial* lower bounds based on the (r, k) -hyperclique hypothesis (albeit with slightly worse exponents compared to the combinatorial ones).

As previously noted, several parameter regimes of this conditional lower bound are nearly tight with algorithms (both previously known algorithms and new ones).

B. Approximation algorithms

a) CYCLEESTIMATIONDIJKSTRA data structure for ANSC.: Algorithms for approximating distances and for approximating Girth, generally have the following structure: Run Dijkstra's algorithm from a random sample of vertices, and

run a *truncated* version of Dijkstra's algorithm from a large set of vertices. For ANSC, we can employ a similar strategy, however the situation becomes slightly more complicated. This is because when we perform Dijkstra's algorithm for ANSC and we detect a cycle, we would like to update the estimate of $SC(v)$ for *all* vertices v on the cycle. To accomplish this, we employ a data structure that we call CYCLEESTIMATIONDIJKSTRA, which uses a modified version of Dijkstra's algorithm along with the power of *link-cut trees* [50] to keep track of the relevant cycle information for all vertices.

Our warm-up algorithm for ANSC that gives a 2-approximation in time $\tilde{O}(mn^{1/2})$ is simply a combination of the CYCLEESTIMATIONDIJKSTRA data structure with the above standard sampling and truncated Dijkstra techniques.

b) Time/accuracy trade-off for better running time.: We first focus on algorithms that achieve better than $\tilde{O}(mn^{1/2})$ running time and worse than 2-approximation. Such a result for n -PSP follows from the following observation: a 2-approximation algorithm in $\tilde{O}(mn^{1/2})$ time can essentially be plugged into the base case of Thorup-Zwick distance oracles [27]. This result for n -PSP appears in the appendix, however for ANSC, we obtain better results and we focus on those here.

Specifically, for ANSC, we obtain a $(k + \varepsilon)$ -approximation in $\tilde{O}(mn^{1/k})$ time (Theorem I.4). A very recent result gave an algorithm for Girth with a similar guarantee. However, our techniques are completely different from theirs. Instead of taking inspiration from an undirected Girth algorithm, we take inspiration from a *directed* Girth algorithm, even though our graph is undirected and techniques for Girth have traditionally been very different in the directed and undirected settings. In general it is not trivial to extend an algorithm for finding cycles from the directed case to the undirected case, as finding undirected cycles introduces challenges that don't appear in directed graphs. To illustrate this, in undirected graphs if we are not careful our algorithm might estimate traversing a path from a node v to u and back to v as a cycle, whereas in the directed case this does not happen. As it turns out, our CYCLEESTIMATIONDIJKSTRA data structure is useful in addressing this issue.

To obtain our $(k + \varepsilon)$ -approximation algorithm for undirected ANSC, we use CYCLEESTIMATIONDIJKSTRA together with a labeling procedure similar to our directed ANSC algorithm, which is in turn from the Girth approximation algorithm of [30]. These techniques allow us to moderate the size of the vertex sets visited while performing Dijkstra's algorithm and prevent over-processing nodes. Specifically, whenever we do CYCLEESTIMATIONDIJKSTRA, we only visit nodes with a particular label, and we change the label of a node v when we know that we must have a good enough estimate for $SC(v)$.

As previously described, finding cycles in undirected graphs presents challenges that are not present for directed graphs, however the opposite is also true; neither setting is clearly strictly harder than the other. We take advantage of the undirected setting to simplify some aspects of the algorithm, which actually yields better bounds for the undirected setting

than the directed setting. In particular, for directed graphs, the labeling procedure is used on top of an induction, however we determine that this induction is unnecessary for undirected graphs, and removing it yields an algorithm with better running time and approximation factor.

c) Time/accuracy trade-off for better approximation ratio.: On the other side of the time/accuracy trade-off, we consider getting a better than 2-approximation with running time slower than $\tilde{O}(mn^{1/2})$. Such a result was previously known for n -PSP [15]. To get such a result for ANSC, we take inspiration from the algorithm of Dahlgaard, Bæk Tejs Knudsen, and Stöckel [51] for approximating the girth of a graph. An algorithm very similar to [51] carries over from Girth to ANSC. The only qualitative differences are our use of edge sampling instead of vertex sampling and our use of the CYCLEESTIMATIONDIJKSTRA data structure.

1) Algorithms in $\tilde{O}(m + n^{2-\varepsilon})$ time: The approximation algorithms we have mentioned so far have time complexities $m \cdot n^c$ for some $c > 0$, which are not desirable for very dense graphs. In our next results, our goal is to minimize the dependency of the running times on m , while keeping them subquadratic in terms of n , that is, we want time complexity $\tilde{O}(m + n^{2-\varepsilon})$ for constant $\varepsilon > 0$.

One simple idea is to use spanners to reduce the number of edges to $m' = n^{2-\Omega(1)}$ (while preserving the distances up to some factor), and then apply our previous algorithms to the sparsified graph in $m' \cdot n^c \ll n^2$ time. This idea has been used e.g. in the distance oracle of Wulff-Nilsen [23]. There are many spanner constructions that take only $\tilde{O}(m)$ time, allowing us to obtain $\tilde{O}(m + n^{2-\varepsilon})$ overall time complexity. This simple idea works for the n -PSP problem, but yields quite a large approximation factor, which is the product of the approximation factors of the spanner and the approximation algorithm. For the ANSC problem, this simple idea does not immediately work, since spanners do not give any guarantees on cycle lengths.

n-PSP.: We briefly explain the idea behind the $(2 + \varepsilon, \beta)$ -approximation algorithm for n -PSP in $\tilde{O}(m + n^{3/2})$ time for any constant $\varepsilon > 0$ and constant β depending on ε (Theorem I.3). We note that this approximation guarantee is nearly tight with our conditional lower bound in Corollary I.1).

We take an $O(n^{3/2})$ -edge subgraph containing the incident edges of all vertices with degree at most $n^{1/2}$, and compute a $(1 + \varepsilon, \beta)$ spanner of size $n^{1+\varepsilon}$ [52] for this subgraph. Then, we perform the $\tilde{O}(m\sqrt{n})$ -time 2-approximation algorithm on this spanner.

Now, it remains to take care of the pairs (s_i, t_i) whose shortest paths pass through some high-degree ($\geq n^{1/2}$) vertex. To do this, we take a sample S of \sqrt{n} nodes, and compute single-source shortest paths from all $s \in S$ in a $(2 + \varepsilon, \beta)$ -spanner of the graph, and use $\min_{x \in S} \{d(s_i, x) + d(x, t_i)\}$ as a $(2 + \varepsilon, \beta')$ estimate of $d(s_i, t_i)$.

Highlight: ANSC.: As mentioned above, spanners do not provide direct guarantees on cycle lengths. However, we observe that they do give some indirect guarantees as follows. Consider a shortest cycle through v , denoted C_v . If we divide

C_v into at least three almost equal subpaths where two of the subpaths have v as an endpoint, each subpath is a shortest path. Take one subpath P . In any k -spanner, there is a subpath P' of length at most $k|P|$ between its endpoints. Since none of these subpaths pass through v except for the ones that start or finish at v , the concatenation of all these approximated subpaths contains a cycle that includes v of length roughly k times $SC(v)$.

Even with the above idea, there are still two issues to overcome for ANSC that are not present for n -PSP: First, when the cycle is very small (length at most 4) we can't apply the above idea. Second, if our spanner has t -additive error and we divide C_v into s subpaths, then the additive error in the estimated cycle length is $t \cdot s$ rather than t .

Our solution to avoid the above issues is to use a *fault-tolerant spanner*. First, we use a 1-fault-tolerant k -spanner, which is a subgraph H such that for each u, v and edge e , if d is the distance between u and v in $G \setminus e$, then $d \leq d_H(u, v) \leq kd$.

The following observation illustrates the usefulness of fault-tolerant spanners for this application. If P is a 1-fault tolerant k -spanner, then for each node v , the subgraph consisting of P and all the edges adjacent to v contains a cycle around v of length at most $k \cdot SC(v)$.

The way we utilize the above observation is as follows. We obtain a sample set S of the nodes, and we perform CYCLEESTIMATIONDIJKSTRA(s) from each $s \in S$ in a subgraph G_s of the underlying graph G . The spanner P is contained in all of the subgraphs G_s , and these subgraphs are selected in a way that for each node v , all the edges adjacent to v appear in at least one subgraph G_s . Moreover, any edge that is not in the spanner P appears in at most 2 of these subgraphs. This means that from the observation above we get an estimate for $SC(v)$ for each v from one of the CYCLEESTIMATIONDIJKSTRAs, and our running time is $O(|S| \cdot |E(P)| + m)$. By using a 1-fault-tolerant 5-spanner, this idea gives us a $(6, 1)$ -approximation algorithm for ANSC in time $\tilde{O}(m + n^{2-1/6})$.

Now, our goal is to obtain a better multiplicative approximation factor. To develop our $(2 + \varepsilon, \beta)$ -approximation algorithm, we first use 1-fault tolerant k -spanners for large k to get approximation algorithms with running time close to linear. We use this algorithm for estimating small (constant sized) cycles. For bigger cycles, instead of fault tolerant spanners, we use the composition of the spanners by [53] and [52], together with the observations mentioned at the beginning of this section.

REFERENCES

- [1] M. Dalirrooyfard, C. Jin, V. Vassilevska Williams, and N. Wein, "Approximation algorithms and hardness for n -pairs shortest paths and all-nodes shortest cycles," *arXiv preprint arXiv:2204.03076*, 2022.
- [2] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani, "Fast estimation of diameter and shortest paths (without matrix multiplication)," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1167–1181, 1999. [Online]. Available: <https://doi.org/10.1137/S0097539796303421>
- [3] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg, "Near-linear time construction of sparse neighborhood covers," *SIAM J. Comput.*, vol. 28, no. 1, pp. 263–277, 1998. [Online]. Available: <https://doi.org/10.1137/S0097539794271898>
- [4] E. Cohen, "Fast algorithms for constructing t -spanners and paths with stretch t ," *SIAM J. Comput.*, vol. 28, no. 1, pp. 210–236, 1998. [Online]. Available: <https://doi.org/10.1137/S0097539794261295>
- [5] D. Coppersmith and M. Elkin, "Sparse sourcewise and pairwise distance preservers," *SIAM J. Discret. Math.*, vol. 20, no. 2, pp. 463–501, 2006. [Online]. Available: <https://doi.org/10.1137/050630696>
- [6] A. Abboud and G. Bodwin, "Reachability preservers: New extremal bounds and approximation algorithms," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, A. Czumaj, Ed. SIAM, 2018, pp. 1865–1883. [Online]. Available: <https://doi.org/10.1137/1.9781611975031.122>
- [7] R. Seidel, "On the all-pairs-shortest-path problem in unweighted undirected graphs," *J. Comput. Syst. Sci.*, vol. 51, no. 3, pp. 400–403, 1995. [Online]. Available: <https://doi.org/10.1006/jcss.1995.1078>
- [8] J. Alman and V. Vassilevska Williams, "A refined laser method and faster matrix multiplication," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2021, pp. 522–539.
- [9] U. Zwick, "All pairs shortest paths using bridging sets and rectangular matrix multiplication," *Journal of the ACM (JACM)*, vol. 49, no. 3, pp. 289–317, 2002.
- [10] F. Le Gall and F. Urrutia, "Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, A. Czumaj, Ed. SIAM, 2018, pp. 1029–1046.
- [11] H. Liu, Z. Zhang, J. Srivastava, and V. Firoiu, "Pwave: A multi-source multi-sink anycast routing framework for wireless sensor networks," in *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, 6th International IFIP-TC6 Networking Conference, Atlanta, GA, USA, May 14-18, 2007, Proceedings*, ser. Lecture Notes in Computer Science, I. F. Akyildiz, R. Sivakumar, E. Ekici, J. C. de Oliveira, and J. McNair, Eds., vol. 4479. Springer, 2007, pp. 179–190. [Online]. Available: https://doi.org/10.1007/978-3-540-72606-7_16
- [12] S. Knopp, P. Sanders, D. Schultes, F. Schulz, and D. Wagner, "Computing many-to-many shortest paths using highway hierarchies," in *Proceedings of the Nine Workshop on Algorithm Engineering and Experiments, ALENEX 2007, New Orleans, Louisiana, USA, January 6, 2007*. SIAM, 2007. [Online]. Available: <https://doi.org/10.1137/1.9781611972870.4>
- [13] L. Roditty and V. Vassilevska Williams, "Fast approximation algorithms for the diameter and radius of sparse graphs," in *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds. ACM, 2013, pp. 515–524. [Online]. Available: <https://doi.org/10.1145/2488608.2488673>
- [14] M. Dalirrooyfard, R. Li, and V. Vassilevska Williams, "Hardness of approximate diameter: Now for undirected graphs," in *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021, pp. 1021–1032. [Online]. Available: <https://doi.org/10.1109/FOCS52979.2021.00102>
- [15] R. Agarwal, "The space-stretch-time tradeoff in distance oracles," in *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, ser. Lecture Notes in Computer Science, A. S. Schulz and D. Wagner, Eds., vol. 8737. Springer, 2014, pp. 49–60. [Online]. Available: https://doi.org/10.1007/978-3-662-44777-2_5
- [16] S. Chechik and T. Zhang, "Nearly 2-approximate distance oracles in subquadratic time," in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2022, pp. 551–580. [Online]. Available: <https://pubs.siam.org/doi/abs/10.1137/1.9781611977073.26>
- [17] R. Yuster, "A shortest cycle for each vertex of a graph," *Inf. Process. Lett.*, vol. 111, no. 21-22, pp. 1057–1061, 2011. [Online]. Available: <https://doi.org/10.1016/j.ipl.2011.07.019>
- [18] P. Sankowski and K. Węgrzycki, "Improved distance queries and cycle counting by frobenius normal form," *Theory Comput. Syst.*, vol. 63, no. 5, pp. 1049–1067, 2019. [Online]. Available: <https://doi.org/10.1007/s00224-018-9894-x>

[19] U. Agarwal and V. Ramachandran, "Fine-grained complexity for sparse graphs," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, I. Diakonikolas, D. Kempe, and M. Henzinger, Eds. ACM, 2018, pp. 239–252. [Online]. Available: <https://doi.org/10.1145/3188745.3188888>

[20] S. Chechik and M. Nechushtan, "Simplifying and unifying replacement paths algorithms in weighted directed graphs," in *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, ser. LIPIcs, A. Czumaj, A. Dawar, and E. Merelli, Eds., vol. 168. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 29:1–29:12. [Online]. Available: <https://doi.org/10.4230/LIPIcs.ICALP2020.29>

[21] A. Shoshan and U. Zwick, "All pairs shortest paths in undirected graphs with integer weights," in *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*. IEEE, 1999, pp. 605–614.

[22] K. Bringmann, M. Künemann, and K. Węgrzycki, "Approximating apsp without scaling: equivalence of approximate min-plus and exact min-max," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 943–954.

[23] C. Wulff-Nilsen, "Approximate distance oracles with improved preprocessing time," in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 202–208.

[24] R. Yuster, "Approximate shortest paths in weighted graphs," *Journal of Computer and System Sciences*, vol. 78, no. 2, pp. 632–637, 2012.

[25] T. M. Chan, "All-pairs shortest paths for real-weighted undirected graphs with small additive error," in *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[26] L. Roditty and A. Shapira, "All-pairs shortest paths with a sublinear additive error," *ACM Transactions on Algorithms (TALG)*, vol. 7, no. 4, pp. 1–12, 2011.

[27] M. Thorup and U. Zwick, "Approximate distance oracles," *Journal of the ACM (JACM)*, vol. 52, no. 1, pp. 1–24, 2005.

[28] M. Patrascu, L. Roditty, and M. Thorup, "A new infinity of distance oracles for sparse graphs," in *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 2012, pp. 738–747.

[29] M. Patrascu and L. Roditty, "Distance oracles beyond the thorup-zwick bound," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010, pp. 815–823.

[30] M. Dalirrooyfard and V. V. Williams, "Conditionally optimal approximation algorithms for the girth of a directed graph," in *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, ser. LIPIcs, A. Czumaj, A. Dawar, and E. Merelli, Eds., vol. 168. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 35:1–35:20. [Online]. Available: <https://doi.org/10.4230/LIPIcs.ICALP2020.35>

[31] A. Lincoln and N. Vyas, "Algorithms and lower bounds for cycles and walks: Small space and sparse graphs," in *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, ser. LIPIcs, T. Vidick, Ed., vol. 151. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 11:1–11:17. [Online]. Available: <https://doi.org/10.4230/LIPIcs.ITCS.2020.11>

[32] K. Hanauer, M. Henzinger, and Q. C. Hua, "Fully Dynamic Four-Vertex Subgraph Counting," in *1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Aspnes and O. Michail, Eds., vol. 221. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 18:1–18:17. [Online]. Available: <https://drops.dagstuhl.de/opus/volltexte/2022/15960>

[33] K. Bringmann, P. Gawrychowski, S. Mozes, and O. Weimann, "Tree edit distance cannot be computed in strongly subcubic time (unless APSP can)," *ACM Trans. Algorithms*, vol. 16, no. 4, pp. 48:1–48:22, 2020. [Online]. Available: <https://doi.org/10.1145/3381878>

[34] A. Abboud, A. Backurs, and V. Vassilevska Williams, "If the current clique algorithms are optimal, so is valiant's parser," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, V. Guruswami, Ed. IEEE Computer Society, 2015, pp. 98–117. [Online]. Available: <https://doi.org/10.1109/FOCS.2015.16>

[35] A. Lincoln, V. Vassilevska Williams, and R. R. Williams, "Tight hardness for shortest cycles and paths in sparse graphs," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, 2018, pp. 1236–1252.

[36] T. Bergamaschi, M. Henzinger, M. P. Gutenberg, V. Vassilevska Williams, and N. Wein, "New techniques and fine-grained hardness for dynamic near-additive spanners," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2021, pp. 1836–1855.

[37] N. Carmeli, N. Tziavelis, W. Gatterbauer, B. Kimelfeld, and M. Riedewald, "Tractable orders for direct access to ranked answers of conjunctive queries," in *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, L. Libkin, R. Pichler, and P. Guagliardi, Eds. ACM, 2021, pp. 325–341. [Online]. Available: <https://doi.org/10.1145/3452021.3458331>

[38] N. Carmeli and M. Kröll, "Enumeration complexity of conjunctive queries with functional dependencies," *Theory of Computing Systems*, vol. 64, no. 5, pp. 828–860, 2020.

[39] N. Carmeli and M. Kröll, "On the enumeration complexity of unions of conjunctive queries," *ACM Trans. Database Syst.*, vol. 46, no. 2, pp. 5:1–5:41, 2021. [Online]. Available: <https://doi.org/10.1145/3450263>

[40] K. Bringmann, N. Fischer, and M. Künemann, "A fine-grained analogue of schaefer's theorem in P: dichotomy of exists^k-forall-quantified first-order graph properties," in *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, ser. LIPIcs, A. Shpilka, Ed., vol. 137. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 31:1–31:27. [Online]. Available: <https://doi.org/10.4230/LIPIcs.CCC.2019.31>

[41] K. Bringmann and J. Slusallek, "Current algorithms for detecting subgraphs of bounded treewidth are probably optimal," in *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, ser. LIPIcs, N. Bansal, E. Merelli, and J. Worrell, Eds., vol. 198. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 40:1–40:16. [Online]. Available: <https://doi.org/10.4230/LIPIcs.ICALP.2021.40>

[42] N. Carmeli, S. Zeevi, C. Berkholz, B. Kimelfeld, and N. Schweikardt, "Answering (unions of) conjunctive queries using random access and random-order enumeration," *CoRR*, vol. abs/1912.10704, 2019. [Online]. Available: <http://arxiv.org/abs/1912.10704>

[43] M. Künemann and D. Marx, "Finding small satisfying assignments faster than brute force: A fine-grained perspective into boolean constraint satisfaction," in *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, ser. LIPIcs, S. Saraf, Ed., vol. 169. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 27:1–27:28. [Online]. Available: <https://doi.org/10.4230/LIPIcs.CCC.2020.27>

[44] V. Vassilevska Williams and Y. Xu, "Truly subcubic min-plus product for less structured matrices, with applications," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, S. Chawla, Ed. SIAM, 2020, pp. 12–29. [Online]. Available: <https://doi.org/10.1137/1.9781611975994.2>

[45] A. Durand, "Fine-grained complexity analysis of queries: From decision to counting and enumeration," in *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, D. Suciu, Y. Tao, and Z. Wei, Eds. ACM, 2020, pp. 331–346. [Online]. Available: <https://doi.org/10.1145/3375395.3389130>

[46] A. Abboud, K. Bringmann, H. Dell, and J. Nederlof, "More consequences of falsifying SETH and the orthogonal vectors conjecture," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, I. Diakonikolas, D. Kempe, and M. Henzinger, Eds. ACM, 2018, pp. 253–266. [Online]. Available: <https://doi.org/10.1145/3188745.3188938>

[47] A. Abboud, K. Bringmann, S. Khoury, and O. Zamir, "Hardness of approximation in p via short cycle removal: Cycle detection, distance oracles, and beyond," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2022)*, 2022, pp. 1487–1500. [Online]. Available: <https://doi.org/10.1145/3519935.3520066>

[48] J. van den Brand, Y. T. Lee, D. Nanongkai, R. Peng, T. Saranurak, A. Sidford, Z. Song, and D. Wang, "Bipartite matching in nearly-linear time on moderately dense graphs," in *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. IEEE, 2020, pp. 919–930. [Online]. Available: <https://doi.org/10.1109/FOCS46700.2020.00090>

- [49] A. Kadria, L. Roditty, A. Sidford, V. Vassilevska Williams, and U. Zwick, “Algorithmic trade-offs for girth approximation in undirected graphs,” in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022.
- [50] D. D. Sleator and R. E. Tarjan, “A data structure for dynamic trees,” *J. Comput. Syst. Sci.*, vol. 26, no. 3, pp. 362–391, 1983. [Online]. Available: [https://doi.org/10.1016/0022-0000\(83\)90006-5](https://doi.org/10.1016/0022-0000(83)90006-5)
- [51] S. Dahlgaard, M. B. T. Knudsen, and M. Stöckel, “New subquadratic approximation algorithms for the girth,” *arXiv preprint arXiv:1704.02178*, 2017.
- [52] M. Thorup and U. Zwick, “Spanners and emulators with sublinear distance errors,” in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 2006, pp. 802–809.
- [53] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie, “Additive spanners and (α, β) -spanners,” *ACM Transactions on Algorithms (TALG)*, vol. 7, no. 1, pp. 1–26, 2010.