Inapproximability of Treewidth and Related Problems (Extended Abstract) *

Yu (Ledell) Wu †, **Per Austrin** ‡, **Toniann Pitassi** and **David Liu**
Department of Computer Science,
University of Toronto, Ontario, Canada
{wuyu,austrin,toni,liudavid}@cs.toronto.edu

Abstract

Graphical models, such as Bayesian Networks and Markov networks play an important role in artificial intelligence and machine learning. Inference is a central problem to be solved on these networks. This, and other problems on these graph models are often known to be hard to solve in general, but tractable on graphs with bounded Treewidth. Therefore, finding or approximating the Treewidth of a graph is a fundamental problem related to inference in graphical models. In this paper, we study the approximability of a number of graph problems: Treewidth and Pathwidth of graphs, Minimum Fill-In, and a variety of different graph layout problems such as Minimum Cut Linear Arrangement. We show that, assuming Small Set Expansion Conjecture, all of these problems are NP-hard to approximate to within any constant factor in polynomial time.

This paper is an extended abstract of the Journal of Artificial Intelligence Research [Wu *et al.*, 2014]

1 Introduction

Graphical models provide a computational framework for efficiently manipulating probability distributions over high dimensional spaces, often involving hundreds of thousands of variables. This framework has found applications in an enormous range of domains including: medical and fault diagnosis, image understanding, speech recognition, web search, coding theory, and statistical physics [Koller and Friedman, 2009]. A graphical model is an efficient representation of a joint distribution over some set of n random variables. Even if the random variables are binary, it is well known that an arbitrary joint distribution requires the specification of 2^n probabilities. Luckily, in the real world, there is often structure in the distribution that allows one to express it more succinctly. A graphical model represents such a joint probability distribution by a graph where the vertices represent the random variables, and the dependences are modeled by the graph structure. Associated with each vertex of the graph is a conditional probability table, which specifies the conditional probabilities of this random variable, conditioned on its neighboring vertices. The two most common types of graphical models are Bayesian networks (also called belief networks), where the underlying graph is directed, and Markov networks (also called Markov random fields), where the underlying graph is undirected. The most basic problem in graphical models is the *inference problem*, which is the problem of computing the posterior marginal distribution of a variable at some vertex. Unfortunately, inference in general is well-known to be NP-hard to compute exactly as well as to approximate [Roth, 1996].

Despite this intractability, an important class of *bounded Treewidth* instances of probabilistic inference has been identified and shown to be exactly computable in polynomial time. The Treewidth of a graph [Robertson and Seymour, 1984] is a fundamental parameter of a graph that measures how close the graph is to being a tree. Treewidth is very closely related to the other notions in machine learning such as Branch-width, Clique-width and Elimination-width (for an overview of Treewidth and related notions, see [Bodlaender *et al.*, 1995]). On graphs with small Treewidth and where the tree decomposition is known, a dynamic programming algorithm yields a polynomial-time algorithm. Particular algorithms for probabilistic inference on bounded Treewidth graphs are the junction-tree method, variable elimination and clique trees (e.g. see [Koller and Friedman, 2009], ch. 9, 10).

The same ideas also yield polynomial-time algorithms and often even linear time algorithms for small Treewidth instances for an astonishing variety of other NP-hard problems, including: satisfiability, counting satisfying assignments, constraint satisfaction, vertex cover, maximum independent set, Hamiltonian circuit, matrix decomposition, and more generally all problems definable in monadic second-order logic. (See the excellent survey [Bodlaender, 2005] for motivation, including theoretical as well as practical applications of Treewidth.) One catch is that for all of these problems, the algorithm must begin by finding a tree decomposition, and then use the decomposition to solve the problem.

*This paper is an extended abstract of the Journal of Artificial Intelligence Research [Wu et al., 2014]. Research supported by NSERC.

†Present Affiliation: Facebook AI Research, 770 Broadway, New York, USA.

‡Present Affiliation: School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden

Given the tree decomposition, the algorithm is typically exponential in the width of the underlying tree decomposition. Thus there is a need for efficient algorithms to actually compute the Treewidth of a given graph, and to find tree decompositions with optimal or close to optimal width.

Unfortunately, while there are many good heuristics for finding a good tree decomposition, it is NP-hard in general to determine the Treewidth of a graph [Arnborg *et al.*, 1987]. However, Bodlaender et al. [Bodlaender *et al.*, 1995] obtained an $O(\log n)$ factor approximation algorithm for Treewidth. In fact, they actually show that if there is a factor c approximation algorithm for vertex separator, then there is an $O(c)$ approximation algorithm for Treewidth. And if there is a factor b approximation algorithm for Treewidth then there is an $O(b \log n)$ approximation algorithm for the related Pathwidth problem. The best currently known approximation factor for vertex separator is $O(\sqrt{\log n})$ [Feige *et al.*, 2005] and thus the best algorithm for Treewidth finds a tree decomposition that is within an $O(\sqrt{\log n})$ factor of the optimal width, and an $O((\sqrt{\log n})(\log n))$ factor approximation algorithm for Pathwidth.

It is a longstanding open question whether or not there is a *constant* factor approximation algorithm for Treewidth. Such an algorithm would lead to faster algorithms to find good tree-decompositions for all of the problems mentioned above. The current best known algorithm that achieves a constant factor approximation for Treewidth runs in time $2^{O(w)}O(n)$, where w is the Treewidth of the underlying graph, and achieves a factor 5 approximation [Bodlaender, 2007]. Similarly, the approximability of many related graph layout problems is also unresolved, including Minimum Cut Linear Arrangement and Interval Graph Completion.

In this paper, we make an important step to resolve this problem by showing that Treewidth, Pathwidth, and a host of related graph layout problems are hard to approximate to within any constant factor, under the Small Set Expansion (SSE) conjecture [Raghavendra and Steurer, 2010].

The SSE conjecture is a strengthened version of the conjecture that P is different from NP and warrants some explanation. In the next subsection (Section 1.1), we explain the SSE conjecture, and how it relates to the P versus NP question and to related conjectures. We then state our main hardness results for Treewidth, Pathwidth and graph layout problems (Sections 1.2, and 1.4), and discuss related results in Section 1.5.

1.1 The Small Set Expansion Conjecture

The P versus NP problem is the most important and intriguing open problem in the field of computational complexity theory. Many decision problems in theory and practice have been proven to be NP-hard, which indicates that they are impossible to compute in polynomial time, under the widely believed conjecture that P $\neq$ NP. The discovery of the PCP theorem in the late 80's [Arora *et al.*, 1998] made it possible to prove that for many optimization problems, approximating the optimal value to within a certain factor is as hard as computing the exact optimal value. Celebrated results show that it is NP-hard to approximate MAX-3SAT within a ratio of $\frac{7}{8} + \epsilon$ for any $\epsilon > 0$ [Håstad, 2001], which gives the optimal lower bound, since there is a simple algorithm that achieves an approximation ratio of $\frac{7}{8}$. Despite this success, for many important problems, such as Vertex Cover, Max-Cut, and Kernel Clustering, the hardness of approximation results obtained through the PCP theorem have not matched the best approximation algorithms known.

The formulation of the Unique Games Conjecture (UGC) due to Khot [Khot, 2002] was intended to clarify the approximability of many optimization problems. The conjecture postulates that the problem of determining the value of a certain type of game, known as a unique game, is NP-hard. Under UGC, many of the known algorithms in approximation are proven to be tight (for an excellent survey on this topic, see [Khot and Vishnoi, 2005]). For instance, under the UGC, the Vertex Cover problem is NP-hard to approximate within a factor of $2 - \epsilon$, for any $\epsilon > 0$ [Khot and Regev, 2008]. Perhaps most strikingly, Raghavendra [Raghavendra, 2008] proved that under the UGC, the semi-definite programming (SDP) approximation algorithm for a large class of constraint satisfaction problems (CSP) are essentially the best one can hope for. Thus, the UGC has become the central open problem in inapproximability and encapsulates the barrier of designing better polynomial time approximation algorithms for a large class of problems.

Despite this tremendous progress, still there remain important yet stubborn problems such as Treewidth, Balanced Separator, Minimum Linear Arrangement (MLA), and many other graph layout problems whose approximation status remains unresolved even assuming the UGC. In the work of Raghavendra and Steurer [Raghavendra and Steurer, 2010], the Small Set Expansion (SSE) Conjecture was introduced, and it was shown that it implies the UGC, and that the SSE Conjecture follows if one assumes that the UGC is true for somewhat expanding graphs. In follow-up work by Raghavendra et al. [Raghavendra *et al.*, 2012], it was shown that the SSE Conjecture is in fact equivalent to the UGC on somewhat expanding graphs, and that the SSE Conjecture implies NP-hardness of approximation for balanced separator and MLA. In this light, the Small Set Expansion conjecture serves as a natural unified conjecture that yields all of the implications of UGC and also hardness for expansion-like problems that could not be resolved with the UGC.

Our main contribution in this paper is to prove that a wide range of other graph layout problems are SSE-hard to approximate to within any constant factor. For these problems, no evidence of hardness of approximation was known prior to our results. Moreover, we show that Treewidth, Pathwidth and Minimum Fill-In are SSE-hard to approximate within any constant factor. This is the first result giving hardness of (relative) approximation for these problems, and gives evidence that no constant factor approximation algorithm exists for them.

It should be noted that the status of the SSE conjecture is very open at this point. In particular, recent results [Arora *et al.*, 2010; Barak *et al.*, 2011; Guruswami and Sinop, 2011] give subexponential-time algorithms for small set expansion. Still despite this recent progress providing evidence against the SSE conjecture, it remains open. Our SSE-hardness results for Treewidth and related problems may therefore be viewed as establishing a new connection between a fundamental conjecture in complexity theory, and the approximability of a ubiquitous problem in artificial intelligence.

1.2 Width Parameters of Graphs

As mentioned earlier, determining the exact Treewidth of a graph and producing an associated optimal tree decomposition is known to be NP-hard [Arnborg *et al.*, 1987], and a central open problem is to determine whether or not there exists a polynomial time constant factor approximation algorithm for Treewidth (see e.g., [Bodlaender *et al.*, 1995; Bodlaender, 2005]). The current best polynomial time approximation algorithm for Treewidth [Feige *et al.*, 2005], computes the Treewidth $\mathsf{tw}(G)$ within a factor $O(\sqrt{\log \mathsf{tw}(G)})$. On the other hand, the only hardness result to date for Treewidth shows that it is NP-hard to compute Treewidth within an *additive* error of n^ϵ for some $\epsilon > 0$ [Bodlaender *et al.*, 1995]. No hardness of approximation is known. In many important special classes of graphs, such as planar graphs [Seymour and Thomas, 1994], and H-minor-free graphs [Feige *et al.*, 2005], constant factor approximations are known, but the general case has remained elusive.

On the positive side, there is a large body of literature developing fixed-parameter algorithms for Treewidth. Exactly determining the Treewidth is fixed-parameter tractable: there is a linear time algorithm that runs in time $2^{poly(k)}poly(n)$ for computing the (exact) Treewidth for graphs of constant treewidth [Bodlaender, 1996]. Constant factor approximation algorithms achieve better dependence on the treewidth, k, and n, with the best such algorithm running in time $2^{O(k)}O(n)$ [Bodlaender, 2007].

A related graph parameter is the so-called *Pathwidth*, which can be viewed as measuring how close G is to a path. The Pathwidth $\mathsf{pw}(G)$ is always at least $\mathsf{tw}(G)$, but can be much larger. The current state of affairs here is similar as for Treewidth; though the current best approximation algorithm only has an approximation ratio of $O(\sqrt{\log \mathsf{pw}(G)} \log n)$ [Feige *et al.*, 2005], the best hardness result is NP-hardness of additive n^ϵ error approximation.

Using the *Small Set Expansion* (SSE) Conjecture [Raghavendra and Steurer, 2010], we show that both $\mathsf{tw}(G)$ and $\mathsf{pw}(G)$ are hard to approximate within any constant factor. In fact, we show something stronger: it is hard to distinguish graphs with small Pathwidth from graphs with large Treewidth. Specifically:

Theorem 1.1. *For every $\alpha > 1$ there is a $c > 0$ such that given a graph $G = (V, E)$ it is SSE-hard to distinguish between the case when $\mathsf{pw}(G) \leq c \cdot |V|$ and the case when $\mathsf{tw}(G) \geq \alpha \cdot c \cdot |V|$.*

In particular, both Treewidth and Pathwidth are SSE-hard to approximate within any constant factor.

1.3 Minimum Fill-In

A closely related graph theoretic property is the Minimum Fill-In of a graph, the minimum number of edges required to add to a graph to triangulate it (i.e., make it chordal). This property has important applications with sparse matrix computations (and in particular Gaussian elimination) and artificial intelligence (see the excellent survey by Heggernes [Heggernes, 2006]).

MINIMUM FILL-IN has been known to be fixed parameter tractable since 1994, when Kaplan et al. [Kaplan *et al.*, 1994] gave an $O(|E|16^k)$ algorithm, where k is the number of edges required. From there, several improvements to the running time have been given, with the most recent in 2012 by Fomin and Villanger 2012, who gave the first subexponential parameterized algorithm, running in time $O(2^{O(\sqrt{k}\log k)} + k^2|V| \cdot |E|)$. In the work of Natanzon et al. 1998, a polynomial time approximation algorithm was presented, which computed a value at most $8k^2$, where k is the optimal solution. For graphs with degree bounded by d, their algorithm achieves an approximation ratio of $O(d^{2.5} \log^4(kd))$.

This remains the best polynomial time approximation algorithm known to date. In particular, it has remained an open question whether a polynomial time constant factor approximation algorithm exists. In this paper, we show that this is not possible, assuming the SSE Conjecture.

Theorem 1.2. *It is SSE-hard to approximate the Minimum Fill-In of a graph to within a constant factor.*

1.4 The Connection: Layout Problems

In a graph layout problem (also known as an arrangement problem, or a vertex ordering problem), the goal is to find an ordering of the vertices, optimizing some condition on the edges, such as adjacent pairs being close. Layout problems are an important class of problems that have applications in many areas such as VLSI circuit design.

A classic example is the *Minimum Cut Linear Arrangement* problem (MCLA). In this problem, the objective is to find a permutation π of the vertices V of an undirected graph $G = (V, E)$, such that the largest number of edges crossing any point,

$$\max_i |\{(u, v) \in E | \pi(u) \leq i < \pi(v)\}|, \qquad (1)$$

is minimized. MCLA is closely related to the *Minimum Linear Arrangement* problem (MLA), in which the max in (1) is replaced by a sum.

The MCLA problem can be approximated to within a factor $O(\log n \sqrt{\log n})$. To the best of our knowledge, there is no hardness of approximation for MCLA in the literature. Its cousin MLA was recently proved SSE-hard to approximate within any constant factor [Raghavendra *et al.*, 2012], and we observe that the same hardness applies to the MCLA problem.

Theorem 1.3. *It is SSE-hard to approximate the Minimum Cut Linear Arrangement problem within any constant factor.*

Another example of graph layout is the *Interval Graph Completion* Problem (IGC). In this problem, the objective is to find a supergraph $G' = (V, E')$ of G with the same vertex set V, such that G' is an interval graph (i.e., the intersection graph of a set of intervals on the real line) and having minimum number of edges. While not immediately appearing to be a layout problem, using a simple structural characterization of interval graphs one can show that IGC can be reformulated as finding a permutation of the vertices that minimizes the sum over the longest edges going out from each vertex, i.e., minimizing

$$\sum_{u \in V} \max_{(u,v) \in E} \max\{\pi(v) - \pi(u), 0\}. \quad (2)$$

See e.g., [Charikar *et al.*, 2010]. The current best approximation algorithm for IGC achieves a ratio of $O(\sqrt{\log n} \log \log n)$ [Charikar *et al.*, 2010]. It turns out that the SSE Conjecture can be used to prove super-constant hardness for this problem as well.

Theorem 1.4. *It is SSE-hard to approximate the Interval Graph Completion problem within any constant factor.*

Theorems 1.3 and 1.4 are just two examples of layout problems that we prove hardness of approximation for. By varying the precise objective function and considering directed acyclic graphs, in which case the permutation π must be a topological ordering of the graph, one can obtain a variety of graph layout problems. We consider a set of eight such problems, generated by three natural variations, and show super-constant SSE-based hardness for all of them in a unified way.

1.5 Previous Work

As the reader may have noticed, for all the problems mentioned, the best current algorithms achieve similar polylogarithmic approximation ratios. Given their close relation, this is of course not surprising. Most of the algorithms are obtained by recursively applying some algorithm for the c-balanced separator problem, in which the objective is to find a bipartition of the vertices of a graph such that both sides contain at least a c fraction of vertices, and the number of edges crossing the partition is minimized.

In the pioneering work on separators by Leighton and Rao [Leighton and Rao, 1999], an $O(\log n)$ approximation algorithm for c-balanced separator was given, which was used to design $O(\log^2 n)$ approximation algorithm for a number of graph layout problems such as MLA, MCLA, and Register Sufficiency. In the groundbreaking work of Arora et al. [Arora *et al.*, 2009], semidefinite programming was used to give an improved approximation ratio of $O(\sqrt{\log n})$ for c-balanced separator. Using their ideas, improved algorithms for ordering problems have been found, such as the $O(\sqrt{\log n} \log \log n)$ approximation algorithm for IGC and MLA [Charikar *et al.*, 2010], the $O(\sqrt{\log n})$ approximation algorithm for Treewidth [Feige *et al.*, 2005] and the $O(\sqrt{\log n} \log n)$ approximation algorithm for Pathwidth [Feige *et al.*, 2005].

On the hardness side, our work builds upon the work of [Raghavendra *et al.*, 2012], which showed that the SSE Conjecture implies superconstant hardness of approximation for MLA (and for c-balanced separator). The only other hardness of relative approximation that we are aware of for these problems is a result of Ambühl et al. [Ambuhl *et al.*, 2007], showing that MLA does not have a PTAS unless NP has randomized subexponential time algorithms.

2 Definitions and Preliminaries

For an undirected graph $G = (V, E)$, and subsets $S, S' \subseteq V$, $E(S, S')$ denotes the set of edges that go between S and S'. In other words, $E(S, S')$ is the set of edges $(u, v) \in E$ such that $u \in S$ and $v \in S'$.

2.1 Treewidth, Elimination Width, and Pathwidth

Definition 2.1 (Tree decomposition, Treewidth). Let $G = (V, E)$ be a graph, T a tree, and let $\mathcal{V} = (V_t)_{t \in T}$ be a family of vertex sets $V_t \subseteq V$ indexed by the vertices t of T. The pair $(T, \mathcal{V})$ is called a *tree decomposition* of G if it satisfies the following three conditions:

(T1) $V = \cup_{t \in T} V_t$;

(T2) for every edge $e \in E$, there exists a $t \in T$ such that both endpoints of e lie in V_t;

(T3) for every vertex $v \in V$, $\{t \in T \mid v \in V_t\}$ is a subtree of T'.

The width of $(T, \mathcal{V})$ is the number $\max\{|V_t| - 1 \mid t \in T\}$, and the *Treewidth* of G, denoted $\mathsf{tw}(G)$, is the minimum width of any tree decomposition of G.

Definition 2.2. Let $G = (V, E)$ be a graph, and let $v_1, \ldots, v_n$ be some ordering of its vertices. Consider the following process: for each vertex v_i in order, add edges to turn the neighborhood of v_i into a clique, and then remove v_i from G. This is an *elimination ordering* of G. The *width* of an elimination ordering is the maximum over all v_i of the degree of v_i when v_i is eliminated. The *elimination width* of G is the minimum width of any elimination order.

Theorem 2.3 (See e.g., [Bodlaender, 2007]). *For every graph G, the elimination width of G equals* $\mathsf{tw}(G)$.

Thus Treewidth is another example of a layout problem. In principle this layout problem can be formulated in the framework of Section 2.3, but the choice of cost function is now more involved than the vertex- and edge-counting considered there.

Definition 2.4 (Path decomposition, Pathwidth). Given a graph G, we say that $(T, \mathcal{V})$ is a *path decomposition* of G if it is a tree decomposition of G and T is a path. The *Pathwidth* of G, denoted $\mathsf{pw}(G)$, is the minimum width of any path decomposition of G.

As claimed earlier, Pathwidth is in fact equivalent with a graph layout problem. (See the next section for the formal definition of layout.)

Theorem 2.5 ([Kinnersley, 1992]). *For every graph G, we have* $\mathsf{pw}(G) = \mathsf{Layout}(G; V, \max)$.

2.2 Minimum Fill-In

Definition 2.6 (Chordal, Triangulation). A graph G is *chordal* if and only if every cycle of length at least 4 has a chord. For any (possibly non-chordal) graph G, a *triangulation* of G is a supergraph of G which is chordal.

Definition 2.7 (Minimum Fill-In). The *Minimum Fill-In* of a graph G is the minimum number of edges required to add to G to triangulate it; i.e., so that the resulting supergraph is chordal.

The problem of determining the Minimum Fill-In of a graph is sometimes called the Chordal Graph Completion problem.

A *perfect elimination ordering* of G is an elimination ordering such that no edges are ever added to G. Put another way, for each vertex v_i, its neighbours appearing after it in the ordering form a clique.

Theorem 2.8 ([Fulkerson and Gross, 1965]). *A graph G is chordal if and only if it has a perfect elimination ordering.*

Treewidth and Minimum Fill-In are related through the following theorem.

Theorem 2.9 (Folklore). *Suppose G is a graph with Treewidth k. Then every triangulation of G has a clique of size $k+1$.*

2.3 Graph Layout Problems

In this subsection, we describe the set of graph layout problems that we consider. A problem from the set is described by three parameters, These three parameters are by no means the only interesting graph layout problems (and some of the settings give rise to more or less uninteresting layout problems). However, they are sufficient to capture the problems we are interested in except Treewidth, which in principle could be incorporated as well though we refrain from doing so in order to keep the definitions simple (see Section 2.1 for more details).

First a word on notation. Throughout the paper, $G = (V, E)$ denotes an undirected graph, and $D = (V, E)$ denotes a directed (acyclic) graph. Letting n denote the number of vertices of the graph, we are interested in bijective mappings $\pi : V \to [n]$. We say that an edge $(u, v) \in E$ *crosses* point $i \in [n]$ (with respect to the permutation π, which will always be clear from context), if $\pi(u) \leq i < \pi(v)$.

We consider the following variations:

1. **Undirected or directed acyclic**: In the case of an undirected graph G, any ordering π of the vertices is a feasible solution. In the case of a DAG D, only the topological orderings of D are feasible solutions.
2. **Counting edges or vertices**: for a point $i \in [n]$ of the ordering, we are interested in the set $E_i(\pi)$ of edges crossing this point. When counting edges, we use the cardinality of E_i as our basic measure. When counting vertices, we only count the set of vertices V_i to the left of i that are incident upon some edge crossing i. In other words, V_i is the projection of $E_i(\pi)$ to the left-hand side vertices. Formally:

   $$E_i(\pi) = \{e \in E \mid \pi(u) \leq i < \pi(v) \text{ where } e = (u, v)\}$$
   $$V_i(\pi) = \{u \in V \mid \pi(u) \leq i < \pi(v) \text{ for some } (u, v) \in E\}$$

   We refer to $|E_i(\pi)|$ or $|V_i(\pi)|$ (depending on whether we are counting edges or vertices) as the *cost* of π at i.
3. **Aggregation by sum or max**: given an ordering π, we aggregate the costs of each point $i \in [n]$, by either summation or by taking the maximum cost.

Given these choices, the objective is to find a feasible ordering π that minimizes the aggregated cost.

Definition 2.10. (Layout value) For a graph H (either an undirected graph G or a DAG D), a cost function C (either E or V), and an aggregation function $\text{agg} : \mathbb{R}^* \to \mathbb{R}$ (either Σ or $\max$), we define $\mathsf{Layout}(H; C, \text{agg})$ as the minimum aggregated cost over all feasible orderings of H. Formally:

$$\mathsf{Layout}(H; C, \text{agg}) = \min_{\text{feasible } \pi} \operatorname*{agg}_{i \in [n]} |C_i(\pi)|.$$

Problem			Equivalent with
undir.	edge	sum	Minimum/Optimal Linear Arrangement
undir.	edge	max	Minimum Cut Linear Arrangement; CutWidth
undir.	vertex	sum	Interval Graph Completion; SumCut
undir.	vertex	max	Pathwidth
DAG	edge	sum	Minimum Storage-Time Sequencing; Directed MLA/OLA
DAG	edge	max	
DAG	vertex	sum	
DAG	vertex	max	Register Sufficiency

Table 1: Taxonomy of Layout Problems

Combining the different choices gives rise to a total of eight layout problems (some more natural than others). Several of these appear in the literature under one or more names, and some turn out to be equivalent[1] to problems that at first sight appear to be different. We summarize some of these names in Table 1. In some cases the standard definitions of these problems look somewhat different than the definition given here (e.g., for Pathwidth, and Interval Graph Completion). For the Pathwidth problem, we discuss these equivalences of definitions in the following two sections.

For Interval Graph Completion, recall from Section 1.4 that the objective is to minimize

$$\sum_{u \in V} \max_{(u,v) \in E} \max\{\pi(v) - \pi(u), 0\}.$$

In other words, we are counting the longest edge going to the right from each point i. If the length of this edge is l then the edge contributes 1 to $V_i(\pi), \ldots, V_{i+l-1}(\pi)$ and hence the objective can be rewritten as

$$\sum_{u \in V} |V_i(\pi)|,$$

so that Interval Graph Completion is $\mathsf{Layout}(G; V, \Sigma)$.

2.4 Small Set Expansion Conjecture

In this section we define the SSE Conjecture. Let $G = (V, E)$ be an undirected d-regular graph. For a set $S \subseteq V$ of vertices, we write $\Phi_G(S)$ for the (normalized) edge expansion of S,

$$\Phi_G(S) = \frac{|E(S, V \setminus S)|}{d|S|}$$

The Small Set Expansion Problem with parameters η and δ, denoted $\mathsf{SSE}(\eta, \delta)$, asks if G has a small set S which does not expand or whether all small sets are highly expanding.

Definition 2.11 ($\mathsf{SSE}(\eta, \delta)$)**.** Given a d-regular graph $G = (V, E)$ [2], $\mathsf{SSE}(\eta, \delta)$ is the problem of distinguishing between the following two cases:

[1] Here, we consider two optimization problems equivalent if there are reductions between them that change the objective values by at most an additive constant.

[2] d is a constant

Yes There is an $S \subseteq V$ with $|S| = \delta|V|$ and $\Phi_G(S) \leq \eta$.

No For every $S \subseteq V$ with $|S| = \delta|V|$ it holds that $\Phi_G(S) \geq 1 - \eta$.

This problem was introduced by Raghavendra and Steurer [Raghavendra and Steurer, 2010], who conjectured that the problem is hard.

Conjecture 2.12 (Small Set Expansion Conjecture). *For every $\eta > 0$, there is a $\delta > 0$ such that $\mathsf{SSE}(\eta, \delta)$ is NP-hard.*

We say that a problem is *SSE-hard* if it is as hard to solve as the SSE problem. Formally, a decision problem $\mathcal{P}$ (e.g., a gap version of some optimization problem) is *SSE-hard* if there is some $\eta > 0$ such that for every $\delta > 0$, $\mathsf{SSE}(\eta, \delta)$ polynomially reduces to $\mathcal{P}$.

Raghavendra et al. 2012 showed that the SSE Problem can be reduced to a quantitatively stronger form of itself. To state this stronger version, we need to first define Gaussian noise stability.

Definition 2.13. Let $\rho \in [-1, 1]$. We define $\Gamma_\rho : [0, 1] \to [0, 1]$ by

$$\Gamma_\rho(\mu) = \Pr\left[X \leq \Phi^{-1}(\mu) \wedge Y \leq \Phi^{-1}(\mu)\right]$$

where Φ^{-1} is inverse function of normal distribution, and X and Y are jointly normal random variables with mean 0 and covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

Fact 2.14. *([Raghavendra* et al.*, 2012]) There is a constant $c > 0$ such that for all sufficiently small ϵ and all $\mu \in [1/10, 1/2]$,*

$$\Gamma_{1-\epsilon}(\mu) \leq \mu(1 - c\sqrt{\epsilon}).$$

Conjecture 2.15 (SSE Conjecture, Equivalent Formulation). *For every integer $q > 0$ and $\epsilon, \gamma > 0$, it is NP-hard to distinguish between the following two cases for a given d-regular graph $G = (V, E)$*

Yes *There is a partition of V into q equi-sized sets $S_1, \ldots, S_q$ such that $\Phi_G(S_i) \leq 2\epsilon$ for every $1 \leq i \leq q$.*

No *For every $S \subseteq V$, letting $\mu = |S|/|V|$, it holds that $\Phi_G(S) \geq 1 - (\Gamma_{1-\epsilon/2}(\mu) + \gamma)/\mu$.*

3 Brief Overview of Reductions

In this section, we give a very brief overview of the reductions used to prove that the layout problems are SSE-hard to approximate within any constant factor.

For the two undirected edge problems (i.e., MLA and MCLA), the hardness follows immediately from the strong form of the SSE Conjecture – for the case of MLA this was proved in [Raghavendra *et al.*, 2012] and the proof for MCLA is similar. This is our starting point for the remaining problems. Unfortunately, the results do not follow from hardness for MLA/MCLA in a black-box way; for the soundness analyses we end up having to use the expansion properties of the original SSE instance.

We then give a reduction from MLA/MCLA with expansion, to the four directed problems. This reduction simply creates the bipartite graph where the vertex set is the union of the edges and vertices of the original graph G, with directed arcs from an edge e to the vertices incident upon e in G. The use of direction here is crucial: it essentially ensures that both the vertex and edge counts of any feasible ordering corresponds very closely to the number of edges crossing the point in the induced ordering of G.

To obtain hardness for the remaining two undirected problems, we perform a similar reduction as for the directed case, creating the bipartite graph of edge-vertex incidences. However, since we are now creating an undirected graph, we can no longer force the edges to be chosen before the vertices upon which they are incident, which was a key property in the reduction for the directed case. In order to overcome this, we duplicate each original vertex a large number of times. This gives huge penalties to orderings which do not "essentially" obey the desired direction of the edges, and makes the reduction work out.

The results for treewidth follows from an additional analysis of the instances produced by the reduction for undirected vertex problems.

Please refer to [Wu *et al.*, 2014] for detailed proofs.

4 Conclusion and Open Problems

We proved SSE-hardness of approximation for a variety of graph problems. Most importantly we obtained the first inapproximability result for the treewidth problem and Minimum Fill-In.

The status of the SSE conjecture is, at this point in time, very uncertain, and our results should therefore not be taken as absolute evidence that there is no polynomial time approximation for (e.g.) treewidth. However, at the very least, our results do give an indication of the difficulty involved in obtaining such an algorithm for treewidth, and builds a connection between these two important problems. We leave the choice of whether to view this as a healthy sign of strength of the SSE Conjecture, or whether to view it as an indication that the conjecture is too strong, to the reader.

There are many important open questions and natural avenues for further work, including:

1. It seems plausible that these results can be extended to a wider range of graph layout problems. For instance, our two choices of aggregators $\max$ and Σ can be viewed as taking ℓ_∞ and ℓ_1 norms, and it seems likely that the results would apply for any ℓ_p norm (though we are not aware of any previous literature studying such variants).
2. It would be nice to obtain hardness of approximation result for our problems based on a weaker hardness assumption such as UGC. It is conjectured in [Raghavendra *et al.*, 2012] that the SSE conjecture is equivalent to UGC. Alternatively, it would be nice to show that hardness of some of our problems imply hardness for the SSE Problem.

References

[Ambuhl *et al.*, 2007] Christoph Ambuhl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability Results for Sparsest Cut, Optimal Linear Arrangement, and Precedence Constrained Scheduling. In *Proceedings of the 48th Annual IEEE Symposium on*

Foundations of Computer Science, FOCS '07, pages 329–337, Washington, DC, USA, 2007. IEEE Computer Society.

[Arnborg *et al.*, 1987] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8:277–284, April 1987.

[Arora *et al.*, 1998] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998.

[Arora *et al.*, 2009] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56:5:1–5:37, April 2009.

[Arora *et al.*, 2010] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 563–572, 2010.

[Barak *et al.*, 2011] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. ECCC Report TR11-065, 2011.

[Bodlaender *et al.*, 1995] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.

[Bodlaender, 1996] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of Small Treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[Bodlaender, 2005] Hans L. Bodlaender. Discovering treewidth. In *SOFSEM*, pages 1–16, 2005.

[Bodlaender, 2007] Hans Bodlaender. Treewidth: Structure and algorithms. In Giuseppe Prencipe and Shmuel Zaks, editors, *Structural Information and Communication Complexity*, volume 4474 of *Lecture Notes in Computer Science*, pages 11–25. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72951-8_3.

[Charikar *et al.*, 2010] Moses Charikar, Mohammad Hajiaghayi, Howard Karloff, and Satish Rao. l_2^2 spreading metrics for vertex ordering problems. *Algorithmica*, 56:577–604, 2010.

[Feige *et al.*, 2005] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum-weight vertex separators. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of computing*, STOC '05, pages 563–572, New York, NY, USA, 2005. ACM.

[Fomin and Villanger, 2012] Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1737–1746. SIAM, 2012.

[Fulkerson and Gross, 1965] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[Guruswami and Sinop, 2011] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for quadratic integer programming with psd objectives, 2011.

[Håstad, 2001] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[Heggernes, 2006] Pinar Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 2006.

[Kaplan *et al.*, 1994] Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping (extended abstract). *SIAM J. Comput*, 28:780–791, 1994.

[Khot and Regev, 2008] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2- ε. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[Khot and Vishnoi, 2005] S. Khot and N.K. Vishnoi. On the unique games conjecture. In *Annual Symposium on Foundations of Computer Science*, volume 46 of *FOCS '05*, page 3. IEEE COMPUTER SOCIETY PRESS, 2005.

[Khot, 2002] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM Symposium on Theory of Computing*, STOC '02, pages 767–775, New York, NY, USA, 2002. ACM.

[Kinnersley, 1992] Nancy G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[Leighton and Rao, 1999] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46:787–832, November 1999.

[Natanzon *et al.*, 1998] Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. In *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing*, STOC '98, pages 41–47, New York, NY, USA, 1998. ACM.

[Raghavendra and Steurer, 2010] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of computing*, STOC '10, pages 755–764, New York, NY, USA, 2010. ACM.

[Raghavendra *et al.*, 2012] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions Between Expansion Problems. IEEE Conference on Computational Complexity, 2012.

[Raghavendra, 2008] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th annual ACM Symposium on Theory of computing*, pages 245–254. ACM, 2008.

[Robertson and Seymour, 1984] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[Roth, 1996] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

[Seymour and Thomas, 1994] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[Wu *et al.*, 2014] Yu Wu, Per Austrin, Toniann Pitassi, and David Liu. Inapproximability of treewidth and related problems. *Journal of Artificial Intelligence Research*, pages 569–600, 2014.