Budgeted Dominating Sets in Uncertain Graphs

Keerti Choudhary
Indian Institute of Technology Delhi, India

Avi Cohen
Tel Aviv University, Israel

N. S. Narayanaswamy
Department of Computer Science and Engineering, IIT Madras, India

David Peleg
Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

R. Vijayaragunathan
Department of Computer Science and Engineering, IIT Madras, India

Abstract

We study the *Budgeted Dominating Set* (BDS) problem on uncertain graphs, namely, graphs with a probability distribution p associated with the edges, such that an edge e exists in the graph with probability $p(e)$. The input to the problem consists of a vertex-weighted uncertain graph $\mathcal{G} = (V, E, p, \omega)$ and an integer *budget* (or *solution size*) k, and the objective is to compute a vertex set S of size k that maximizes the expected total domination (or total weight) of vertices in the closed neighborhood of S. We refer to the problem as the *Probabilistic Budgeted Dominating Set* (PBDS) problem. In this article, we present the following results on the complexity of the PBDS problem.

1. We show that the PBDS problem is NP-complete even when restricted to uncertain *trees* of diameter at most four. This is in sharp contrast with the well-known fact that the BDS problem is solvable in polynomial time in trees. We further show that PBDS is W[1]-hard for the budget parameter k, and under the *Exponential time hypothesis* it cannot be solved in $n^{o(k)}$ time.
2. We show that if one is willing to settle for $(1 - \epsilon)$ approximation, then there exists a PTAS for PBDS on trees. Moreover, for the scenario of uniform edge-probabilities, the problem can be solved optimally in polynomial time.
3. We consider the parameterized complexity of the PBDS problem, and show that Uni-PBDS (where all edge probabilities are identical) is W[1]-hard for the parameter pathwidth. On the other hand, we show that it is FPT in the combined parameters of the budget k and the treewidth.
4. Finally, we extend some of our parameterized results to planar and apex-minor-free graphs.

Our first hardness proof (Thm. 1) makes use of the new problem of k-Subset $\Sigma-\Pi$ Maximization (k-SPM), which we believe is of independent interest. We prove its NP-hardness by a reduction from the well-known k-SUM problem, presenting a close relationship between the two problems.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Uncertain graphs, Dominating set, NP-hard, PTAS, treewidth, planar graph

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.32

Related Version *Full Version*: https://arxiv.org/abs/2107.03020

Acknowledgements We thank an anonymous reviewer for pointing us to [6], yielding a shorter proof of the FPT algorithm for Uni-PBDS parameterized by treewidth and k.
David Peleg is supported by the Venky Harinarayanan and Anand Rajaraman Visiting Chair Professorship at the Indian Institute of Technology Madras, Chennai, India (IIT Madras). Supported by the chair's funds, this work was done in part when David Peleg, Avi Cohen, and Keerti Choudhary visited IIT Madras and when R. Vijayaragunathan visited the Weizmann Institute of Science, Rehovot, Israel.

1 Introduction

Background and Motivation. Many optimization problems in network theory deal with placing resources in key vertices in the network so as to maximize coverage. Some practical contexts where such coverage problems occur include placing mobile towers in wireless networks to maximize reception, assigning emergency vehicle centers in a populated area to guarantee fast response, opening production plants to ensure short distribution lines, and so on. In the context of social networks, the problem of spreading influencers so as to affect as many of the network members as possible has recently attracted considerable interest.

Coverage problems may assume different forms depending on the optimized parameter. A basic "full coverage" variant is the classical *dominating set* problem, which asks to find a *minimum* vertex set S such that each vertex not in S is *dominated by* S, i.e., is adjacent to at least one vertex in S. In the dual *budgeted dominating set (BDS)* problem, given a bound k (the budget), it is required to find a set S of size at most k *maximizing* the number of covered vertices. Over *vertex weighted* graphs, the goal is to maximize the total *weight* of the covered vertices, also known as the *domination*. It is this variant that we're concerned with here.

Traditionally, coverage problems involve a fixed network of static topology. The picture becomes more interesting when the network structure is uncertain, due to potential edge connections and disconnections or link failures. Pre-selection of resource locations at the design stage becomes more challenging in such partial-information settings.

In this work, we study the problem in one of the most fundamental settings, where the input is a graph whose edges fail independently with a given probability. The goal is to find a k-element set that maximizes the *expected* (1-hop) coverage (or domination). Our results reveal that the probabilistic versions of the coverage problem are significantly harder than their deterministic counterparts, and analyzing them require more elaborate techniques.

An *uncertain graph* $\mathcal{G}$ is a triple (V, E, p), where V is a set of n vertices, $E \subseteq V \times V$ is a set of m edges, and the function $p : E \to [0, 1]$ assigns a probability of existence to each edge in E. So an m edge uncertain graph $\mathcal{G}$ represents a probability space consisting of 2^m graphs, sometimes called *possible worlds*, derived by sampling each edge $e \in E$ independently with probability $p(e)$. For $H = (V, E' \subseteq E)$, the event of sampling H as a possible world, denoted $H \sqsubseteq \mathcal{G}$, occurs with probability $\Pr(H \sqsubseteq \mathcal{G}) = \prod_{e \in E'} p(e) \prod_{e \in E \setminus E'} \big(1 - p(e)\big)$. The notion of possible worlds dates back to Leibniz and *possible world semantics (PWS)* is well-studied in the modal logic literature, beginning with the work of Kripke.

Our work focuses on budgeted dominating sets on vertex-weighted uncertain graphs, i.e., the *Probabilistic Budgeted Dominating Set* (PBDS) problem. The input consists of a vertex-weighted uncertain graph $\mathcal{G} = (V, E, p, \omega)$, with a weight function $\omega : V \to \mathbb{Q}^+$ and an integer budget k. Set $p(vv) = 1$ for every v. For a vertex u and a set $S \subseteq V$, denote by $\Pr(u \sim S) = 1 - \prod_{v \in S}(1 - p(uv))$ the probability that $u \in S$ or u is connected to some vertex in S. For sets $S_1, S_2 \subseteq V$, the expected coverage (or domination) of S_1 by S_2 is defined as $\mathcal{C}(S_1, S_2) = \sum_{v \in S_1} \big(w(v) \Pr(v \sim S_2)\big)$. The PBDS problem aims to find a set S of size k that maximizes $\mathcal{C}(V, S)$ over the possible worlds. Its decision version is defined as follows.

Probabilistic budgeted dominating set (PBDS)

Input: A vertex-weighted uncertain graph $\mathcal{G} = (V, E, p, \omega)$, an integer k and a target domination value t.
Question: Is there a set $S \subseteq V$ of size at most k such that $\mathcal{C}(V, S) \geq t$?

Our Results and Discussion. The budgeted dominating set problem is known to have a polynomial time solution on trees. A natural question is if the same applies to the probabilistic version of the problem. We answer this question negatively, showing the following.

▶ **Theorem 1.** *The* PBDS *problem is NP-hard on uncertain trees of diameter* 4*. Furthermore, (i) the* PBDS *problem on uncertain trees is* W[1]*-hard for the parameter* k*, and (ii) an* $n^{o(k)}$ *time solution to* PBDS *will falsify the Exponential time hypothesis.*

In order to prove the theorem, we introduce the following problem.

SUBSET $\Sigma - \Pi$ MAXIMIZATION (k-SPM)

Input: A multiset $\mathcal{A} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ of N pairs of positive rationals, an integer k, and a rational t.

Question: Is there a set $S \subseteq [N]$ of size exactly k satisfying $\sum_{i \in S} x_i - \prod_{i \in S} y_i \geq t$?

To establish the complexity of the k-SPM problem, we present a polynomial time reduction from k-SUM to k-SPM, thereby proving that both k-PBDS and k-SPM are NP-hard. Moreover, Downey and Fellows [23] showed that the k-SUM problem is W[1]-hard, implying that if k-SUM has an FPT solution with parameter k, then the W hierarchy collapses. This provides our second hardness result.

▶ **Theorem 2.** *The* k-SPM *problem is* W[1]*-hard for the parameter* k*. Furthermore, any* $N^{o(k)}$ *time solution to* k-SPM *falsify the Exponential time hypothesis.*

The k-SUM problem can be solved easily in $\widetilde{O}(n^{\lceil k/2 \rceil})$ time. However, it has been a long-standing open problem to obtain any polynomial improvement over this bound [1, 47]. Patrascu and Williams [48] showed an $n^{o(k)}$ time algorithm for k-SUM falsifies the famous *Exponential time hypothesis* (ETH). Hence, our polynomial time reductions also imply that any algorithm optimally solving k-PBDS or k-SPM must require $n^{\Omega(k)}$ time unless ETH fails.

▶ **Theorem 3.** *Under the* k-SUM *conjecture, for any* $\varepsilon > 0$*, there does not exist an* $n^{\lceil k/2 \rceil - \varepsilon}$ *time algorithm to* PBDS *problem on vertex-weighted uncertain trees.*

An intriguing question is whether the k-SPM is substantially harder than k-SUM. For the simple scenario of $k = 2$, the 2-SUM problem has an $O(n \log n)$ time solution. However, it is not immediately clear whether the 2-SPM problem has a truly sub-quadratic time solution (i.e., $O(n^{2-\varepsilon})$ time for some $\varepsilon > 0$). We leave this as an open question. This is especially of interest due to the following result.

▶ **Theorem 4.** *Let* $1 \leqslant c < 2$ *be the smallest real such that* 2-SPM *problem has an* $\widetilde{O}(n^c)$ *time algorithm. Then, there exists an* $\widetilde{O}\big((dn)^{c\lceil k/2 \rceil + 1}\big)$ *time algorithm for optimally solving* k-PBDS *on trees with arbitrary edge-probabilities, for some constant* $d > 0$*.*

Given the hardness of k-PBDS on uncertain trees, it is of interest to develop efficient approximation algorithms. Clearly, the expected neighborhood size of a vertex set is a submodular function, and thus it is known that the greedy algorithm yields a $(1 - 1/\mathrm{e})$-approximation for the PBDS problem in general uncertain graphs [39, 46]. For uncertain *trees*, we improve this by presenting a fully polynomial-time approximation scheme for PBDS.

▶ **Theorem 5.** *For any integer* k*, and any* n*-vertex tree with arbitrary edge probabilities, a* $(1 - \epsilon)$*-approximate solution to the optimal probabilistic budgeted dominating set* (PBDS) *of size* k *can be computed in time* $\widetilde{O}(k^2 \epsilon^{-1} n^2)$*.*

We also consider a special case that the number of distinct probability edges on the input uncertain tree is bounded above by some constant γ.

▶ **Theorem 6.** *For any integer k, and an n-vertex tree $\mathcal{T}$ with at most γ edge probabilities, an optimal solution for the* PBDS *problem on $\mathcal{T}$ can be computed in time $\tilde{O}(k^{(\gamma+2)}n)$.*

We investigate the complexity of PBDS on bounded treewidth graphs. The hardness construction on bounded treewidth graphs is much more challenging. Due to this inherent difficulty, we focus on the *uniform* scenario, where all edge probabilities $p(e)$ are identical. We refer to this version of the problem as *Uni-PBDS*. We show that for any $0 < q < 1$, the Uni-PBDS problem with edge-probability q is W[1]-hard for the pathwidth parameter of the input uncertain graph $\mathcal{G}$. In contrast, the BDS problem (when all probabilities are one) is FPT when parameterized by the pathwidth of the input graph.

▶ **Theorem 7.** *Uni-PBDS is* W[1]*-hard w.r.t. the pathwidth of the input uncertain graph.*

Then, we consider the Uni-PBDS problem with combined k and treewidth parameters. We show that the Uni-PBDS problem can be formulated as a variant of the Extended Monadic Second order (EMS) problem due to Arnborg et al. [6], to derive an FPT algorithm for the Uni-PBDS problem parameterized by the treewidth of $\mathcal{G}$ and k.

▶ **Theorem 8.** *For any integer k, and any n-vertex uncertain graph of treewidth w with uniform edge probabilities, k-Uni-*PBDS *can be solved in time $O(f(k,w)n^2)$, and thus is FPT in the combined parameter involving k and w. By dynamic programming on a nice tree decomposition we can show that $f(k,w)$ is $k^{O(w)}$ (we do not present this here).*

Finally, using the structural property of dominating sets from Fomin et al. [28], we derive FPT algorithms parameterized by the budget k in apex-minor-free graphs and planar graphs.

▶ **Theorem 9.** *For any integer k, and any n-vertex weighted planar or apex-minor free graph, the Uni-PBDS problem can be solved in time $2^{O(\sqrt{k}\log k)}n^{O(1)}$.*

Related Work. Uncertain graphs have been used in the literature to model the uncertainty among relationships in protein-protein interaction networks in bioinformatics [7], road networks [8, 37] and social networks [22, 39, 52, 54]. Connectivity [9, 10, 33, 38, 51, 53], network flows [27, 31], structural-context similarity [55], minimum spanning trees [26], coverage [16, 34, 35, 44, 43], and community detection [11, 49] are well-studied problems on uncertain graphs. In particular, budgeted coverage problems model a wide variety of interesting combinatorial optimization problems on uncertain graphs. For example, the classical facility location problem [36, 40] is a variant of coverage. As another example, in a classical work, Kempe, Kleinberg, and Tardos [39] study influence maximization problem as an expected coverage maximization problem in uncertain graphs. They consider the scenario where influence propagates probabilistically along relationships, under different influence propagation models, like the *Independent Cascade (IC)* and *Linear Threshold (LT)* models, and show that choosing k influencers to maximize the expected influence is NP-hard in the IC model. The coverage problem in the presence of uncertainty was studied extensively also in sensor placement and w.r.t. the placement of light sources in computer vision. SS A special case of the budgeted coverage problem is the *Most Reliable Source (MRS)* problem, where given an uncertain graph $\mathcal{G} = (V, E, p)$, the goal is to find a vertex $u \in V$ such that the expected number of vertices in u's connected component is maximized. To the best of our knowledge, the computational complexity of MRS is not known, but

it is polynomial time solvable on some specific graph classes like trees and series-parallel graphs [12, 19, 20, 21, 43]. Domination is another special kind of coverage and its complexity is very well-studied. The classical dominating set (DS) problem is known to be W[2]-hard in general graphs [23], and on planar graphs it is fixed parameter tractable with respect to the size of the dominating-set as the parameter [32]. Further, on H-minor-free graphs, the dominating-set problem is solvable in subexponential time [4, 17]. It also admits a linear kernel on H-minor-free graphs and graphs of bounded expansion [3, 25, 29, 30, 50]. On graphs of treewidth bounded by w, the classical dynamic programming approach [15] can be applied to show that the DS problem is FPT when parameterized by w. The *Budgeted Dominating Set (BDS)* problem is known to be NP-hard [42] as well as W[1]-hard for the budget parameter [23]. Furthermore, a subexponential parameterized algorithm is known for BDS on apex-minor-free graphs [28]. The treewidth-parameterized FPT algorithm for the dominating-set problem can be adapted to solve the BDS problem in time $O(3^w kn)$. In particular, for trees there exists a linear running time algorithm. PBDS was studied as Max-Exp-Cover-1-RF in the survey paper [45], and given a dynamic programming algorithm on a nice tree decomposition with runtime $2^{O(w\cdot\Delta)}n^{O(1)}$, where Δ is the maximum degree of $\mathcal{G}$. The question whether PBDS has a treewidth parameterized FPT algorithm remained unresolved; it is settled in the negative in this work.

2 Preliminaries

Consider a simple undirected graph $G = (V, E)$ with vertex set V and edge set E, and let $n = |V|$ and $m = |E|$. Given a vertex subset $S \subseteq V$, the subgraph induced by S is denoted by $G[S]$. For a vertex $v \in V$, $N(v)$ denotes the set of neighbors of v and $N[v] = N(v) \cup \{v\}$ is the closed neighborhood of v. Let $deg(v)$ denote the degree of the vertex v in G. A vertex subset $S \subseteq V$ is said to be a *dominating set* of G if every vertex $u \in V \setminus S$ has a neighbor $v \in S$. For an integer $r > 0$, a vertex subset $S \subseteq V$ is said to be an *r-dominating set* of G if for every vertex $u \in V \setminus S$ there exists a vertex $v \in S$ at distance at most r from u. A graph H is said to be an *apex* if it can be made planar by the removal of at most one vertex. A graph G is said to be *apex-minor-free* if it does not contain as its minor some fixed apex graph H. All planar graphs are apex-minor-free as they do not contain as minor the apex graphs $K_{3,3}$ and K_5. The notations $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{N}$ denote, respectively, the sets of real, rational, and natural numbers (including 0). For integers $a \leq b$, define $[a, b]$ to be the set $\{a, a + 1, \ldots, b\}$, and for $b > 0$ let $[b] \equiv [1, b]$. Other than this, we follow standard graph theoretic and parameterized complexity terminology [15, 18, 24].

Tree Decomposition. A *Tree decomposition* of an undirected graph $G = (V, E)$ is a pair $(\mathtt{T}, X)$, where $\mathtt{T}$ is a tree whose set of nodes is $X = \{X_{\mathbf{i}} \subseteq V \mid \mathbf{i} \in V(\mathtt{T})\}$, such that

1. for each edge $u \in V$, there is an $\mathbf{i} \in V(\mathtt{T})$ such that $u \in X_{\mathbf{i}}$,
2. for each edge $uv \in E$, there is an $\mathbf{i} \in V(\mathtt{T})$ such that $u, v \in X_{\mathbf{i}}$, and
3. for each vertex $v \in V$ the set of nodes $\{\mathbf{i} \mid v \in X_{\mathbf{i}}\}$ forms a subtree of $\mathtt{T}$.

The *width* of a tree decomposition $(\mathtt{T}, X)$ equals $\max_{\mathbf{i}\in V(\mathtt{T})} |X_{\mathbf{i}}| - 1$. The treewidth of a graph G is the minimum width over all tree decompositions of G.

A tree decomposition $(\mathtt{T}, X)$ is *nice* if $\mathtt{T}$ is rooted by a node $\mathbf{r}$ with $X_{\mathbf{r}} = \emptyset$ and every node in $\mathtt{T}$ is either an *insert node*, *forget node*, *join node* or *leaf node*. Thereby, a node $\mathbf{i} \in V(\mathtt{T})$ is an insert node if $\mathbf{i}$ has exactly one child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \cup \{v\}$ for some $v \notin X_{\mathbf{j}}$; it is a forget node if $\mathbf{i}$ has exactly one child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \setminus \{v\}$ for some $v \in X_{\mathbf{j}}$; it is a join node if $\mathbf{i}$ has exactly two children $\mathbf{j}$ and $\mathbf{h}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} = X_{\mathbf{h}}$; and it is a leaf node if $X_{\mathbf{i}} = \emptyset$. Given a tree decomposition of width w, a nice tree decomposition of width w and $O(w\ n)$ nodes can be obtained in linear time [41].

A tree decomposition $(\mathcal{T}, \mathcal{X})$ is said to be a *path decomposition* if $\mathcal{T}$ is a path. The *pathwidth* of a graph G is minimum width over all possible path decompositions of G. Let $\text{pw}(G)$ and $\text{tw}(G)$ denote the pathwidth and treewidth of the graph G, respectively. The pathwidth of a graph G is one lesser than the minimum clique number of an interval supergraph H which contains G as an induced subgraph. It is well-known that the maximal cliques of an interval graph can be linearly ordered so that for each vertex, the maximal cliques containing it occur consecutively in the linear order. This gives a path decomposition of the interval graph. A path decomposition of the graph G is the path decomposition of the interval supergraph H which contains G as an induced subgraph. In our proofs we start with the path decomposition of an interval graph and then reason about the path decomposition of graphs that are constructed from it.

Uncertain Graphs. For an uncertain graph $\mathcal{G} = (V, E, p, \omega)$, the vertex weights and the probabilities associated with the edges are given as rationals. The treewidth and diameter of $\mathcal{G}$ are the same as the treewidth and diameter of its maximal possible world, $G = (V, E)$.

Numerical Approximation. When analyzing our polynomial reductions, we employ numerical analysis techniques to bound the error in numbers obtained as products of an exponential and the root of an integer. We use the following well-known bound on the error in approximating an exponential function by the sum of the lower degree terms in the series expansion.

▶ **Lemma 10** ([5]). *For $z \in [-1, 1]$, e^z can be approximated using the Lagrange remainder as*

$$e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \ldots + \frac{z^Q}{Q!} + R_Q(z)$$

where $|R_Q(z)| \leq e/(Q+1)! \leq 1/2^Q$.

We use the following lemma for bounding the error in multiplying approximate values.

▶ **Lemma 11.** *For any set $\{d_1, \ldots, d_k\}$ of k reals in the range $[0, 1]$,*

$$\prod_{i \in [k]} (1 - d_i) \;\geq\; 1 - \sum_{i \in [k]} d_i.$$

Proof. The proof is by induction on k. The base case of $k = 1$ trivially holds. For any two reals $a, b \in [0, 1]$, $(1-a)(1-b) \geq 1 - (a+b)$. Applying this result iteratively yields that for any $k \geq 1$, if $\prod_{i \in [k-1]}(1 - d_i) \geq 1 - (\sum_{i \in [k-1]} d_i)$, then

$$\prod_{i \in [k]} (1 - d_i) \;\geq\; \left(1 - \Big(\sum_{i \in [k-1]} d_i\Big)\right)(1 - d_k) \;\geq\; 1 - \sum_{i \in [k]} d_i.$$

The claim follows. ◀

3 Hardness Results on Trees

3.1 k-SPM hardness

We first show that the k-Subset $\Sigma - \Pi$ Maximization (k-SPM) problem is NP-hard by a reduction from the k-SUM problem. Let $\langle X, k \rangle$ with $X = \{x_1, \ldots, x_N\}$ be an instance of the k-SUM problem. Let $L = 1 + \max_{i \in [N]} |x_i|$.

Denote by $\langle \mathcal{A}, k, t\rangle$ an instance of the k-SPM problem. Given an instance $\langle X, k\rangle$ of k-SUM, we compute the array $\mathcal{A}(X) = \{(\tilde{x}_i, \tilde{y}_i) \mid i \in [N]\}$ of the k-SPM problem as follows. For $1 \leq i \leq N$, set $\tilde{x}_i := (L + x_i)/(kL)$.

Let $Q = 3\log_2(kL)$. For $i \in [N]$, define $y_i = \mathrm{e}^{x_i/(kL)}$, and let $\tilde{y}_i$ be a rational approximation of y_i that is computed using Lemma 10 such that $0 \leq y_i - \tilde{y}_i \leq 1/2^Q$. The new instance of the k-SPM problem is $\langle \mathcal{A}(X), k, t = 0\rangle$.

Observe that for each $i \in [N]$, $\tilde{y}_i \geq y_i - 1/2^Q \geq \mathrm{e}^{-1/k} - 1/(kL)^3 \geq 1/2$, for $k \geq 3$. Thus, the elements of $\mathcal{A}(X)$ are positive rationals. The next lemma provides a crucial property of any set S of vertices of size k.

▶ **Lemma 12.** *Let $\lambda = \frac{1}{(2kL)^2}$. For each $S \subseteq [N]$ of size k, $0 \leq \prod_{i\in S} y_i - \prod_{i\in S} \tilde{y}_i \leq \lambda$.*

Proof. Let $\alpha = 1/(kL)^3$. We have:

$$\begin{aligned}\textstyle\prod_{i\in S} y_i - \prod_{i\in S} \tilde{y}_i \leq\ & \textstyle\prod_{i\in S} y_i - \prod_{i\in S}(y_i - \alpha) \ =\ \prod_{i\in S} y_i\Big(1 - \prod_{i\in S}\big(1 - \frac{\alpha}{y_i}\big)\Big)\\ & \leq\ \textstyle\prod_{i\in S} y_i\big(\sum_{i\in S} \frac{\alpha}{y_i}\big) \ \leq\ \mathrm{e}^{\sum_{i\in S} x_i/(kL)}\, \alpha k\, \mathrm{e}^{1/k} \ \leq\ \alpha k\, \mathrm{e}^2 \ \leq \frac{1}{4(kL)^2}\ ,\end{aligned}$$

where the second inequality is obtained by Lemma 11. The claim follows. ◀

We now establish the correctness of the reduction.

▶ **Theorem 13.** *The k-SUM problem is polynomial-time reducible to k-SPM.*

Proof. Let $M = \sum_{i\in[N]}(x_i + L)$. Define a real valued function $F(z) = z - \mathrm{e}^{-1+z}$ with domain $[0, M/(kL)]$. Observe that $F(1) = 0$ and the derivative is $F'(1) = 0$. The function $F(\cdot)$ is clearly concave, which indicates that:

- **(i)** $F(z) \leq 0$, for each $z \in [0, M/(kL)]$,
- **(ii)** $F(z)$ obtains its unique maximum at $z = 1$, where its value is 0, and
- **(iii)** When restricted to the values in the set $\big\{z/(kL) \mid z \in [0, M] \text{ is an integer}\big\}$, $F(z)$ obtains its second largest value at $z = 1 - 1/(kL)$.

For any $S \subseteq [N]$, denote $z_S = \sum_{i\in S} \tilde{x}_i$. For a set $S \subseteq [N]$ of size k, we have:

$$\begin{aligned}\textstyle\sum_{i\in S} \tilde{x}_i - \prod_{i\in S} y_i \ &=\ z_S - \mathrm{e}^{\sum_{i\in S} x_i/(kL)} \ =\ z_S - \mathrm{e}^{\sum_{i\in S} \tilde{x}_i/(kL) - 1/k} && (1)\\ &=\ z_S - \mathrm{e}^{-1}\cdot \mathrm{e}^{z_S/(kL)} \ =\ F(z_S). && (2)\end{aligned}$$

By combining Lemma 12 and Eq. (2), we obtain the following.

$$F(z_S) \ \leq\ \textstyle\sum_{i\in S} \tilde{x}_i - \prod_{i\notin S} \tilde{y}_i \ \leq\ F(z_S) + \lambda.$$

On the other hand, for any set $S \subseteq [N]$ of size k for which $F(z_S) < 0$, we have $F(z_S) \leq F(1 - 1/(kL)) = (1 - 1/(kL) - e^{-1/(kL)})$.

$$F(z_S) \ \leq\ F(1 - 1/(kL)) \ =\ (1 - 1/(kL) - e^{-1/(kL)}).$$

Further,

$$1 - \tfrac{1}{kL} - e^{-1/(kL)} \ \leq\ \big(1 - \tfrac{1}{kL}\big) - \big(1 - \tfrac{1}{kL} + \tfrac{1}{2(kL)^2} - \tfrac{1}{6(kL)^3}\big) \ \leq\ -\tfrac{1}{4(kL)^2} \ =\ -\lambda\ .$$

So, for a set S, $\sum_{i\in S} \tilde{x}_i - \prod_{i\in S} \tilde{y}_i \geq 0$ if and only if $\sum_{i\in S} \tilde{x}_i = 1$, or equivalently $\sum_{i\in S} x_i = 0$. It follows that $\langle X, k\rangle$ is a yes instance of the k-SUM problem if and only if $\langle \mathcal{A}(X), k, t = 0\rangle$ is a yes instance of the k-SPM problem. The time to compute $\tilde{x}_i$ and $\tilde{y}_i$ from x_i is polynomial in $Q \cdot \log_2(kL)$, for $1 \leq i \leq N$. Thus, the time-complexity of our reduction is $N \cdot \log_2^{O(1)}(kL)$, which is at most polynomial in N as long as $L = 2^{O(N)}$. Hence, the k-SUM problem is polynomial-time reducible to the k-SPM problem. ◀

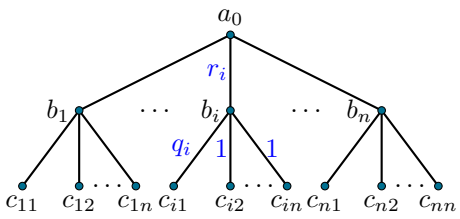

Figure 1 Illustration of the lower bound of Theorem 1. Here $p_i = 1 - y_i/(X_{\max} \cdot Y_{\max})$ and $q_i = x_i/(X_{\max} \cdot Y_{\max})^k$ for $i \in [N]$.

Proof of Theorem 2. The reduction given in the proof of Theorem 13 is a parameter preserving reduction for the parameter k. That is, the parameters in the instances of the k-SUM and the k-SPM problem are same in values and the constructed instance of the k-SPM problem is of size polynomial in the input size of the k-SUM instance. Thus, the reduction preserves the parameter k. Since the k-SUM problem is known to be W[1]-hard for the parameter k [2, 23], the k-SPM problem is also W[1]-hard for the parameter k. Further, it is known that under the Exponential time hypothesis (ETH), there cannot exist an $o(N^k)$ time solution for the k-SUM problem [48], so it follows that under ETH there is no $o(N^k)$ time algorithm for k-SPM as well. ◀

3.2 Hardness of PBDS on Uncertain Trees

In this subsection, we show the hardness results for the PBDS problem on trees, establishing Theorem 1. In order to achieve this, we present a polynomial time reduction from k-SPM to PBDS on unweighted trees.

Proof of Theorem 1. In order to prove our claim, we provide a reduction from k-SPM to PBDS. Given an instance $\langle \mathcal{A} = ((x_1, y_1), \ldots, (x_N, y_N)), k, t \rangle$ of the k-SPM problem, where t is a rational, an equivalent instance of the PBDS problem is constructed as follows. Let $n = N^2 + N + 1$. Construct an uncertain tree $\mathcal{T} = (V, E, p)$, where the vertex set V consists of three disjoint sets, namely, $A = \{a_0\}$, $B = \{b_1, \ldots, b_N\}$, and $C = \{c_{11}, c_{12}, \ldots, c_{NN}\}$ (see Figure 1. Note that the uncertain tree $\mathcal{T}$ is considered to be unweighted or unit weight on the vertices. The vertex a_0 is connected by edges to the vertices in B. For each $1 \leq i \leq n$, the vertex b_i is connected by edges to the vertices $c_{i1} \ldots, c_{iN}$. Let $X_{\max} = \max\{1, x_1, x_2, \ldots, x_N\}$ and $Y_{\max} = \max\{1, y_1, y_2, \ldots, y_N\}$. To complete the construction, define the probability function $p : E \to [0, 1]$ as follows:

$$p(v\bar{v}) = \begin{cases} r_i = 1 - (y_i)/(X_{\max} \cdot Y_{\max}), & \text{if } v\bar{v} = a_0 b_i \text{ for } 1 \leq i \leq N, \\ q_i = x_i/(X_{\max} \cdot Y_{\max})^k, & \text{if } v\bar{v} = b_i c_{i1} \text{ for } 1 \leq i \leq N, \\ 1, & \text{otherwise.} \end{cases}$$

Since $x_i, y_i \gneq 0$ for each $1 \leq i \leq n$, we have that $p(v, \bar{v}) \in [0, 1]$ is rational for every $(v, \bar{v}) \in E$. This completes the construction of the instance for the PBDS problem. We show that the given instance $\langle \mathcal{A}, k, t \rangle$ is a yes instance of k-SPM if and only if $\mathcal{T}$ has a set S of size k such that $\mathcal{C}(V, S) \geq 1 + (N-1)k + t/(X_{\max} Y_{\max})^k$.

Let S_{opt} be a set of size k maximizing $\mathcal{C}(V, S_{opt})$ in $\mathcal{T}$. We show $S_{opt} \subseteq B$. Assume, to the contrary, that there exists some $z \in S_{opt}$ satisfying $z \notin B$. Consider $i \in [N]$ such that none of the vertices $c_{i1}, \ldots, c_{iN}$ lie in S_{opt}. Such i must exist since $|S_{opt}| = k$. If $z \in C$, then replacing z by b_i results in a set $S' = S_{opt} \setminus \{z\} \cup \{b_i\}$ such that $\mathcal{C}(V, S') \geq \mathcal{C}(V, S_{opt}) + N - 3$,

contradicting the optimality of S_{opt}. Hence, S_{opt} must be contained in $A \cup B$. In this case, z must be a_0. Now the set $S' = S_{opt} \setminus \{z\} \cup \{b_i\}$ is such that $\mathcal{C}(V, S') \geq \mathcal{C}(V, S_{opt}) + N/2 - 2$. Since $N \geq 6$, this contradicts the optimality of S_{opt}. It follows that $S_{opt} \subseteq B$.

Next, consider a set $S \subseteq B$ of size k, and let $I_S = \{i \in [N] \mid b_i \in S\}$. We have

$$\begin{aligned}\mathcal{C}(V,S) &= \left(1 - \textstyle\prod_{i\in I_S}(1 - p(a_0, b_i))\right) + \left(\sum_{i\in I_S}(p(b_i, c_{i1}) + N - 1)\right)\\ &= 1 + (N-1)k \ + \textstyle\sum_{i\in I_S} x_i/(X_{\max}\cdot Y_{\max})^k \ - \prod_{i\in I_S} y_i/(X_{\max}\cdot Y_{\max})\\ &= 1 + (N-1)k \ + (X_{\max}\cdot Y_{\max})^{-k}\left(\textstyle\sum_{i\in I_S} x_i \ - \prod_{i\in I_S} y_i\right) \ .\end{aligned}$$

This formulation of the coverage function shows that the given instance $\langle \mathcal{A}, k, t\rangle$ is a yes instance of k-SPM if and only if $\mathcal{T}$ has a set S of vertices of size k such that $\mathcal{C}(V, S) \geq 1 + (N-1)k + t/(X_{\max}Y_{\max})^k$. Thus, the k-SPM problem is reduced in polynomial time to PBDS on unweighed trees.

It remains to prove NP-hardness, W[1]-hardness, and $n^{o(k)}$ lower-bound under ETH. Note that the above reduction is a parameterized preserving reduction for the parameter k. That is, the parameter k in the k-SPM problem is the solution size (also called k) parameter for the PBDS problem. Since the k-SPM problem (i) is NP-hard, (ii) is W[1]-hard for the parameter k, and (iii) cannot have time complexity $n^{o(k)}$ under the Exponential time hypothesis (by Theorem 2), it follows that the same hardness results hold for PBDS as well. Therefore, the PBDS problem on uncertain trees (i) is NP-hard, (ii) is W[1]-hard for the parameter k, and (iii) cannot have time complexity $n^{o(k)}$ if ETH holds true. ◀

The k-SUM conjecture [1, 47] states that the k-SUM, for the parameters N and k, requires at least $N^{\lceil k/2\rceil - o(1)}$ time.

▶ **Conjecture 14** (k-SUM Conjecture). *There do not exist a $k \geq 2$, an $\varepsilon > 0$, and an algorithm that succeeds (with high probability) in solving k-SUM in $N^{\lceil k/2\rceil - \varepsilon}$ time.*

Proof of Theorem 3. Consider the uncertain tree $\mathcal{T}$ constructed in the proof of Theorem 1. We set $n_0 = 0$. Modify the original construction of $\mathcal{T}$ by deleting the $N - 2$ vertices: $c_{i3}, c_{i4}, \ldots, c_{iN}$, and setting $\omega_{c_{i2}} = N - 1$, for $1 \leq i \leq N$. Thus the tree contains exactly $n = 3N + 1$ vertices. Now, k-SUM is reducible to k-SPM, and k-SPM is reducible to PBDS, both in polynomial time, and, moreover the parameter k remains unaltered and the size of problem grows by at most constant factor. This shows that, for $\varepsilon > 0$, an $n^{\lceil k/2\rceil - \varepsilon}$ time algorithm to weighted PBDS implies an $N^{\lceil k/2\rceil - \Omega(\varepsilon)}$ time algorithm to k-SUM, thereby, falsifying the k-SUM conjecture. ◀

Note that since the PBDS problem is NP-hard on trees, it is also para-NP-hard [15, 24] for the treewidth parameter.

4 Hardness of Uni-PBDS for the pathwidth parameter

In this section, we show that even for the restricted case of uniform probabilities, the Uni-PBDS problem is W[1]-hard for the pathwidth parameter, and thus also for treewidth (Theorem 7). This is shown by a reduction from the Multi-Colored Clique problem to the Uni-PBDS problem. It is well-known that the Multi-Colored Clique problem is W[1]-hard for the parameter solution size [23].

MULTI-COLORED CLIQUE
Input: A positive integer k and a k-colored graph G.
Parameter: k
Question: Does there exist a clique of size k with one vertex from each color class?

Let $(G = (V, E), k)$ be an input instance of the MULTI-COLORED CLIQUE problem, with n vertices and m edges. Let $V = (V_1, \ldots, V_k)$ denote the partition of the vertex set V in the input instance. We assume, without loss of generality, $|V_i| = n$ for each $i \in [k]$. For each $1 \leq i \leq k$, let $V_i = \{u_{i,\ell} \mid 1 \leq \ell \leq n\}$.

4.1 Gadget based reduction from Multi-Colored Clique

Let (G, k) be an instance of the MULTI-COLORED CLIQUE problem. For any probability $0 < p < 1$, and for any integer f such that $f > \max\{knm, n + k^2/p\}$, our reduction constructs an uncertain graph $\mathcal{G}$. The output of the reduction is an instance $(\mathcal{G}, k', t')$ of the Uni-PBDS problem where each edge has probability p, $k' = (n+1)(m+kn)$ and $t' = (kn+m)\big((n+1)fp + n + np + 1 + 2(1-(1-p)^n)\big) + 4\binom{k}{2}(1-(1-p)^{n+1})$. In the presentation below, we show that this choice of k' and t' ensures that there is a set of size k' with expected domination at least t' in $\mathcal{G}$ if and only if G has a multi-colored clique of size k.

We first construct a gadget graph to represent the vertices and edges of the input instance of the MULTI-COLORED CLIQUE problem. We construct two types of gadgets, $\mathcal{D}$ and $\mathcal{I}$ in the reduction, illustrated in Figure 2 (in Appendix A.1). The gadget $\mathcal{I}$ is the primary gadget and $\mathcal{D}$ is a secondary gadget used to construct $\mathcal{I}$. When we refer to a gadget, we mean the primary gadget $\mathcal{I}$ unless the gadget $\mathcal{D}$ is specified. For each vertex and edge in the given graph, our reduction has a corresponding gadget. The gadget $\mathcal{D}$ is defined as follows.

Gadget of type $\mathcal{D}$. Given a pair of vertices u and v, the gadget $\mathcal{D}_{u,v}$ consists of vertices u, v, and f additional vertices. The vertices u and v are made adjacent to every other vertex. We refer to the vertices u and v as *heads*, and remaining vertices of $\mathcal{D}_{u,v}$ as *tails*, and u are v are said to be connected by the gadget $\mathcal{D}_{u,v}$.

▶ **Observation 15.** *The pathwidth of a gadget of type $\mathcal{D}$ is 2.*

Gadget of type $\mathcal{I}$. We begin the construction of the gadget with $2n$ vertices partitioned into two sets where each partition contains n vertices. Let $A = \{a_1, \ldots, a_n\}$ and $C = \{c_1, \ldots, c_n\}$ be this partition. For each $i \in [n]$, vertices a_i and c_i are connected by the gadget $\mathcal{D}_{a_i,c_i}$. Let h_a and h_c be two additional vertices connected by the gadget $\mathcal{D}_{h_a,h_c}$. The vertices in the sets A and C are made adjacent to h_a and h_c, respectively. This completes the construction of the gadget. In the reduction, a gadget of type $\mathcal{I}$ is denoted by the symbol $\mathcal{I}$ along with an appropriate subscript based on whether the gadget is associated with a vertex or an edge.

▷ Claim 16. The pathwidth of a gadget of type $\mathcal{I}$ is at most 4.

Description of the reduction. For $1 \leq i < j \leq k$, let $E_{i,j} = \{xy \mid x \in V_i, y \in V_j\}$ be the set of edges with one end point in V_i and the other in V_j in G. For each $1 \leq i < j \leq k$, the graph $\mathcal{G}$ has an induced subgraph $\mathcal{G}_i$ corresponding to V_i, and has an induced subgraph $\mathcal{G}_{i,j}$ for the edge set $E_{i,j}$. We refer to $\mathcal{G}_i$ as a vertex-partition block and $\mathcal{G}_{i,j}$ as an edge-partition block. Inside block $\mathcal{G}_i$, there is a gadget of type $\mathcal{I}$ for each vertex in V_i, and in the block $\mathcal{G}_{i,j}$, there is a gadget for each edge in $E_{i,j}$. For a vertex $u_{i,x}$, $\mathcal{I}_x$ denotes the gadget corresponding to $u_{i,x}$ in the partition V_i, and for an edge e, $\mathcal{I}_e$ denotes the gadget corresponding to e. The blocks are appropriately connected by connector vertices which are defined below.

We start by defining the structure of a block denoted by B. The definition of the block applies to both the vertex-partition block and the edge-partition block. A block B consists of gadgets and additional vertices as follows (See Figure 3 in Appendix A.1).

- The block B corresponding to the vertex-partition block $\mathcal{G}_i$ for any $i \in [k]$ is described as follows: for each $\ell \in [n]$, add a gadget $\mathcal{I}_\ell$ to the vertex-partition block $\mathcal{G}_i$, to represent the vertex $u_{i,\ell} \in V_i$. In addition to the gadgets, we add $n+1$ vertices to the block B described as follows: Let $F(B) = \{b_1, \ldots, b_n, d_i\}$ be the set of additional vertices that are added to the block B. For each $\ell \in [n]$, the vertices in the set C of the gadget $\mathcal{I}_\ell$ in the block B are made adjacent to b_ℓ. For each $\ell \in [n]$, the vertices in the set A of the gadget $\mathcal{I}_\ell$ in the block B are made adjacent to d_i.
- The block B corresponding to the edge-partition block $\mathcal{G}_{i,j}$ for any $1 \leq i < j \leq k$ is described as follows: for each $e \in E_{i,j}$, add a gadget $\mathcal{I}_e$ in the edge-partition block $\mathcal{G}_{i,j}$, to represent the edge e. In addition to the gadgets, we add $|E_{i,j}|+1$ vertices to the block B described as follows: Let $F(B) = \{b_e \mid e \in E_{i,j}\} \cup \{d_{i,j}\}$ be the set of additional vertices that are added to the block B. For each $e \in E_{i,j}$, the vertices in the set C of the gadget $\mathcal{I}_e$ in the block B are made adjacent to b_e. For each $e \in E_{i,j}$, the vertices in the set A of the gadget $\mathcal{I}_e$ in the block B are made adjacent to $d_{i,j}$.

The blocks defined above are connected by the connector vertices described next. These connector vertices are used to connect the edge-partition blocks and vertex-partition blocks, and thus ensure that each edge in G is appropriately represented in $\mathcal{G}$. Let $R = \{r^i_{i,j}, s^i_{i,j}, r^j_{i,j}, s^j_{i,j} \mid 1 \leq i < j \leq k\}$ be the connector vertices. The blocks are connected based on the cases described below. The connections involving the $\mathcal{I}$ gadgets in two vertex-partition blocks and an $\mathcal{I}$ gadget in an edge-partition block is illustrated in Figure 4 in Appendix A.1. First, we describe the connection of vertex-partition blocks corresponding V_i and V_j to the appropriate connector vertices. Following this, we describe the connection of the two vertex-partition blocks to the edge-partition block corresponding to $E_{i,j}$ through the appropriate connector vertices.

For each $i \in [k]$, each $i < j \leq k$ and each $\ell \in [n]$,

- for each $1 \leq t \leq \ell$, a_t in the gadget $\mathcal{I}_\ell$ of $\mathcal{G}_i$ is made adjacent to $s^i_{i,j}$, and
- for each $\ell \leq t \leq n$, a_t in the gadget $\mathcal{I}_\ell$ of $\mathcal{G}_i$ is made adjacent to the vertex $r^i_{i,j}$.

For each $i \in [k]$, each $1 \leq j < i$ and each $\ell \in [n]$,

- for each $1 \leq t \leq \ell$, a_t in the gadget $\mathcal{I}_\ell$ of $\mathcal{G}_i$ is made adjacent to the vertex $s^i_{j,i}$, and
- for each $\ell \leq t \leq n$, a_t in the gadget $\mathcal{I}_\ell$ of $\mathcal{G}_i$ is made adjacent to the vertex $r^i_{j,i}$.

Now, we describe the edges to connect the $\mathcal{I}$ gadgets in the vertex-partition blocks $\mathcal{G}_i$ and $\mathcal{G}_j$ and to the appropriate $\mathcal{I}$ gadgets in the edge-partition block $\mathcal{G}_{i,j}$. For each $1 \leq i < j \leq k$, and for each $e = u_{i,x}u_{j,y} \in E_{i,j}$,

- for each $1 \leq t \leq x$, a_t in the gadget $\mathcal{I}_e$ of $\mathcal{G}_{i,j}$ is made adjacent to the vertex $r^i_{i,j}$, and
- for each $x \leq t \leq n$, a_t in the gadget $\mathcal{I}_e$ of $\mathcal{G}_{i,j}$ is made adjacent to the vertex $s^i_{i,j}$.
- for each $1 \leq t \leq y$, a_t in the gadget $\mathcal{I}_e$ of $\mathcal{G}_{i,j}$ is made adjacent to the vertex $r^j_{i,j}$, and
- for each $y \leq t \leq n$, a_t in the gadget $\mathcal{I}_e$ of $\mathcal{G}_{i,j}$ is made adjacent to the vertex $s^j_{i,j}$.

This completes the construction of the graph $\mathcal{G}$ with $O(mn^2)$ vertices and $O(mn^3)$ edges.

▷ Claim 17. The pathwidth of a block B is at most 6.

The following lemma bounds the pathwidth of the graph $\mathcal{G}$ by a polynomial in k.

▶ **Lemma 18.** *The pathwidth of the graph $\mathcal{G}$ is at most $4\binom{k}{2} + 6$.*

Due to space constraints, we have deferred the complete proof to the expanded version.

Proof of Theorem 7. Given an instance (G, k) of Multi-Colored Clique , the instance $(\mathcal{G}, k')$ is constructed in polynomial time where k' and t' are polynomial in input size. By Lemma 18, the pathwidth of $\mathcal{G}$ is a quadratic function of k. Finally, we can also show that the Uni-PBDS instance $(\mathcal{G}, k', t')$ output by the reduction is equivalent to the Multi-Colored Clique instance (G, k) that was input to the reduction. Since Multi-Colored Clique is known to be W[1]-hard for the parameter k, it follows that the Uni-PBDS problem is W[1]-hard with respect to the pathwidth parameter of the input graph. ◀

This completes the proof of Theorem 7.

5 PBDS on Trees: PTAS and Exact Algorithm

In this section, we present our algorithmic results for the PBDS problem on trees. Throughout this section, assume $\mathcal{T}$ is rooted at some vertex r. For each $x \in V$, denote by $\textsc{par}(x)$ the parent of x in V, and by $\mathcal{T}(x)$ the subtree of $\mathcal{T}$ rooted at x.

5.1 PTAS for PBDS on Trees

For each $v \in V$ and each $b \in [0, k]$, define $\mathcal{Y}_v(par, curr, b)$ to be the optimal value of $\mathcal{C}(V(\mathcal{T}(v)), S)$ where par and $curr$ are boolean indicator variables that, respectively, denote whether or not $\textsc{par}(v)$ and v are in S (written below as $I_{\textsc{par}(v)\in S}$ and $I_{v\in S}$), and b denotes the number of descendants of v in S. Formally, $\mathcal{Y}_v(par, curr, b)$ is represented as follows:

$$\arg\max \left\{ \textstyle\sum_{x\in\mathcal{T}(v)} \mathcal{C}(x, S) \;\middle|\; S \subseteq V,\ |S \cap (\mathcal{T}(v) \setminus v)| = b,\ curr = I_{v\in S},\ par = I_{\textsc{par}(v)\in S} \right\}$$

The main idea behind our PTAS is to use the rounding method. Instead of computing $\mathcal{Y}_x$, we compute its approximation, represented as $\widehat{\mathcal{Y}}_x$. This is done in a bottom-up fashion, starting from leaf nodes of $\mathcal{T}$. For each $x \in V$, define $\delta(x)$ to be $|\mathcal{Y}_x - \widehat{\mathcal{Y}}_x|$. Throughout our algorithm, we maintain the invariant that $\widehat{\mathcal{Y}}_x \leq \mathcal{Y}_x$, for every $x \in V$.

We now present an algorithm to compute $\widehat{\mathcal{Y}}$. Since $\mathcal{Y}_x$ is easy to compute for a leaf x, we set $\widehat{\mathcal{Y}}_x = \mathcal{Y}_x$. For a leaf x, $\mathcal{Y}_x(par, curr, b)$ is (i) undefined if $b \neq 0$, (ii) ω_x if $curr = 1, b = 0$, (iii) $\omega_x\ p_{(\textsc{par}(x),x)}$ if $par = 1, curr = 0, b = 0$, and (iv) zero otherwise. Consider a non-leaf v. Let $z_1, \ldots, z_t$ be v's children in $\mathcal{T}$, and z_0 be v's parent in $\mathcal{T}$ (if exists). Let $\mathcal{L}(\beta)$, for $\beta \geq 0$, denote the collection of all integral vectors $\sigma = (b_1, curr_1, \ldots, b_t, curr_t)$ of length $2t$ satisfying (i) $curr_i \in \{0, 1\}$ and $b_i \geq 0$, for $i \in [1, t]$, and (ii) $\sum_{i\in[1,t]}(b_i + curr_i) = \beta$. In our representation of σ as $(b_1, curr_1, \ldots, b_t, curr_t)$, the term $curr_i$ corresponds to the indicator variable representing whether or not z_i lies in our tentative set S, and b_i corresponds to the cardinality of $S \cap \big(V(\mathcal{T}(z_i)) \setminus z_i\big)$. Further, for $i \in [1, t]$, let $\mathcal{L}_i(\beta)$ be the collection of those vectors $\sigma = (b_1, curr_1, \ldots, b_t, curr_t) \in \mathcal{L}(\beta)$ that satisfy $b_j, curr_j = 0$ for $j > i$.

For a given $curr, par, b \geq 0$, we now explain the computation of $\widehat{\mathcal{Y}}_v(par, curr, b)$. Assume that we have already computed the approximate values $\widehat{\mathcal{Y}}_{z_i}$ $(i \in [1, t])$ corresponding to v's children in $\mathcal{T}$. Setting $W = \max_{u\in V} \omega_u$, and using the scaling factor $M = \epsilon W/n$, let

$$A(\sigma) = \begin{cases} \omega_v, & \text{if curr=1,} \\ M \left\lfloor \frac{\omega_v}{M} \Big(1 - (1 - par \cdot p_{(z_0,v)}) \cdot \prod_{\substack{i\in[1,t] \\ curr_i=1}} (1 - p_{(z_i,v)})\Big) \right\rfloor, & \text{otherwise,} \end{cases} \tag{3}$$

$$B(\sigma) = \sum_{i\in[1,t]} \widehat{\mathcal{Y}}_{z_i}(curr, curr_i, b_i), \tag{4}$$

$$\widehat{\mathcal{Y}}_v(par, curr, b) = \max_{\sigma\in\mathcal{L}(b)} \big(A(\sigma) + B(\sigma)\big). \tag{5}$$

In order to efficiently compute $\widehat{\mathcal{Y}}_v$, we define the notion of *preferable* vectors. For any two vectors $\sigma_1, \sigma_2 \in \mathcal{L}(\beta)$, we say that σ_1 is *preferred* over σ_2 (and write $\sigma_1 \geq \sigma_2$) if both (i) $A(\sigma_1) \geq A(\sigma_2)$, and (ii) $B(\sigma_1) \geq B(\sigma_2)$. For $i \in [1,t]$, let $\mathcal{L}_i^*(\beta)$ be a *maximal* subset of $\mathcal{L}_i(b)$ such that $\sigma_1 \not\geq \sigma_2$ for any two vectors $\sigma_1, \sigma_2 \in \mathcal{L}_i^*(\beta)$.

Define $\phi_v = |\{A(\sigma) \mid \sigma \in \mathcal{L}(\beta), \text{ for } \beta \in [0,k]\}|$. The following observation is immediate by the definition of $\mathcal{L}_i^*$.

▶ **Observation 19.** *For each $i \in [1,t]$ and $\beta \in [0,k]$, $|\mathcal{L}_i^*(\beta)| \leq \phi_v$.*

In order to compute $\widehat{\mathcal{Y}}_v(par, curr, b)$, we explicitly compute and store $\mathcal{L}_i^*(\beta)$, for $1 \leq i \leq t$. The set $\mathcal{L}_1^*(\beta)$ is quite easy to compute. Let $\sigma_1 = (\beta, 0, 0, \ldots, 0)$ and $\sigma_2 = (\beta - 1, 1, 0, \ldots, 0)$ be the only two vectors lying in $\mathcal{L}_1(\beta)$. Then $\mathcal{L}_1^*(\beta)$ is that vector among σ_1 and σ_2 that maximizes the sum $A(\sigma) + B(\sigma)$.

The lemma below provides an iterative procedure for computing the sets $\mathcal{L}_i^*(\beta)$, for $i \geq 2$.

▶ **Lemma 20.** *For every $i, \beta \geq 1$, the set $\mathcal{L}_i^*(\beta)$ can be computed from $\mathcal{L}_{i-1}^*(\beta)$ in time $\widetilde{O}\big(\beta + \sum_{\alpha \in [0,\beta]} |\mathcal{L}_{i-1}^*(\alpha)|\big)$.*

Proof. Initialize $\mathcal{L}_i^*(\beta)$ to $\emptyset$. At each stage, maintain the list $\mathcal{L}_i^*(\beta)$ sorted by the values $A(\cdot)$, and reverse-sorted by the values $B(\cdot)$. Our algorithm to compute $\mathcal{L}_i^*(\beta)$ involves the following steps.

1. For each $curr \in \{0,1\}$ and $b \in [0,\beta]$, first compute a set $\mathcal{P}_{b,curr}$ obtained by replacing the values b_i and $curr_i$ in each $\sigma \in \mathcal{L}_{i-1}^*(\beta - (curr + b))$ by b and $curr$ respectively. Let $\mathcal{P} = \bigcup_{b \in [0,\beta], curr \in \{0,1\}} \mathcal{P}_{b,curr}$.
2. For each $\sigma \in \mathcal{P}$, check in $O(\log |\mathcal{P}|)$ time if there is a $\sigma' \in \mathcal{L}_i^*(\beta)$ that is preferred over σ (i.e. $\sigma' \geq \sigma$). If no such σ' exists, then (a) add σ to $\mathcal{L}_i^*(\beta)$, and (b) remove all those σ'' from $\mathcal{L}_i^*(\beta)$ that are less preferred than σ, that is, $\sigma'' < \sigma$.

The runtime of the algorithm is $O(\beta + |\mathcal{P}| \log |\mathcal{P}|)$ which is at most $\widetilde{O}(\beta + \sum_{\alpha \in [0,\beta]} |\mathcal{L}_{i-1}^*(\alpha)|)$.

Next we now prove its correctness. Consider a $\sigma = (b_1, curr_1, \ldots, b_t, curr_t) \in \mathcal{L}_i(\beta)$. It suffices to show that if $\sigma \notin \mathcal{P}_{b_i,curr_i}$, then there exists a $\sigma' \in \mathcal{P}_{b_i,curr_i}$ satisfying $\sigma' \geq \sigma$. Let σ_0 be obtained from σ by replacing $b_i, curr_i$ with 0. Since $\sigma \notin \mathcal{P}_{b_i,curr_i}$, it follows that $\sigma_0 \notin \mathcal{L}_{i-1}(\beta - (b_i + curr_i))$. So there must exist a vector $\sigma_0' = (b_1', curr_1', \ldots, b_t', curr_t') \in \mathcal{L}_{i-1}(\beta - (b_i + curr_i)$ satisfying $A(\sigma_0') \geq A(\sigma_0)$ and $B(\sigma_0') \geq B(\sigma_0)$. Let σ' be the vector obtained from σ_0' by replacing $b_i', curr_i'$ with $b_i, curr_i$. It can be easily verified from Eq. (3) and (4), that $A(\sigma') \geq A(\sigma)$ and $B(\sigma') \geq B(\sigma)$. Since the constructed σ' indeed lies in $\mathcal{P}_{b_i,curr_i}$, the proof follows. ◀

The following claim is an immediate corollary of Lemma 20.

▶ **Lemma 21.** *The value of $\widehat{\mathcal{Y}}_v(par, curr, b)$, for any $par, curr \in \{0,1\}$ and $b \in [0,k]$, is computable in $\widetilde{O}(b \cdot \deg(v) \cdot \phi_v)$ time, given the values of $\widehat{\mathcal{Y}}_{z_i}$ for $i \leq t$.*

Proof. Observe that $\widehat{\mathcal{Y}}_v(par, curr, b) = \max_{\sigma \in \mathcal{L}_t^*(b)} \big(A(\sigma) + B(\sigma)\big)$, where $A(\sigma)$ and $B(\sigma)$ are as defined in Eq. (3) and (4). By Observation 19 and Lemma 20, the total computation time of the set $\mathcal{L}_t^*(b)$ is at most $\widetilde{O}(b \cdot t \cdot \phi_v)$, which is equal to $\widetilde{O}(b \cdot \deg(v) \cdot \phi_v)$. ◀

Lemma 21 implies that starting from leaf nodes, the values $\widehat{\mathcal{Y}}_x(par, curr, b)$ can be computed in bottom-up manner, for each valid choice of triplet $(par, curr, b)$ and each $x \in V$, in total time $\widetilde{O}(k^2 n \cdot \max_{v \in V} \phi_v)$. We now prove $\phi_v = O(\epsilon^{-1} n)$. If $curr = 1$, then $A(\sigma)$ takes only one value. If $curr = 0$, then the value of $A(\sigma)$ is a multiple of M and is also bounded above by W. This implies that the number of distinct values $A(\sigma)$ can take is indeed bounded by $W/M = O(\epsilon^{-1} n)$.

▶ **Proposition 22.** *Computing $\widehat{\mathcal{Y}}_x$ for all $x \in V$ takes in total $\widetilde{O}(k^2 n \cdot \max_{x\in V} \phi_x) = \widetilde{O}(k^2\epsilon^{-1}n^2)$ time.*

5.2 Approximation Analysis of PTAS on Trees

We provide here the approximation analysis of the $(1-\epsilon)$-bound. Let

$$S_{opt} = \arg\max_{S\subseteq V,|S|=k} \mathcal{C}(V,S) = \arg\max_{S\subseteq V,|S|=k} \sum_{x\in V} \omega_x \Pr(x\sim S),$$
$$\widehat{S}_{opt} = \arg\max_{S\subseteq V,|S|=k} \Big(\sum_{x\in S} \omega_x \Pr(x\sim S) + \sum_{x\in V\setminus S} M\Big\lfloor \frac{\omega_x \Pr(x\sim S)}{M} \Big\rfloor \Big).$$

Observe that $\max\{\mathcal{Y}_r(0,0,k), \mathcal{Y}_r(0,1,k-1)\} = \mathcal{C}(V,S_{opt})$ and $\max\{\widehat{\mathcal{Y}}_r(0,0,k), \widehat{\mathcal{Y}}_r(0,1,k-1)\} = \mathcal{C}(V,\widehat{S}_{opt})$. The following lemma proves that $\widehat{S}_{opt}$ indeed achieves a $(1-\epsilon)$-approximation bound.

▶ **Lemma 23.** $(1-\epsilon)\,\mathcal{C}(V,S_{opt}) \le \mathcal{C}(V,\widehat{S}_{opt}) \le \mathcal{C}(V,S_{opt})$.

Proof. In order to prove the first inequality, we first show that $\mathcal{C}(V,S_{opt}) - \mathcal{C}(V,\widehat{S}_{opt}) \le \epsilon\,\mathcal{C}(V,S_{opt})$.

$$\begin{aligned}\mathcal{C}(V,S_{opt}) - \mathcal{C}(V,\widehat{S}_{opt}) &\le \max_{\substack{S\subseteq V\\|S|=k}} \Big(\sum_{x\in V} \omega_x \Pr(x\sim S) - \sum_{x\in S} \omega_x \Pr(x\sim S) - \sum_{x\in V\setminus S} M\Big\lfloor \frac{\omega_x \Pr(x\sim S)}{M} \Big\rfloor \Big)\\ &\le (n-k)M \le \epsilon\, W \le \epsilon\,\mathcal{C}(V,S_{opt}).\end{aligned}$$

Next, for each $x\in V$ and $S\subseteq V$, we have $M\lfloor M^{-1}\omega_x \Pr(x\sim S)\rfloor \le \omega_x \Pr(x\sim S)$, thereby implying that $\mathcal{C}(V,\widehat{S}_{opt}) \le \mathcal{C}(V,S_{opt})$. This completes our proof. ◀

Proof of Theorem 5. For any integer k, any n-vertex tree $\mathcal{T}$ with arbitrary edge probabilities, and for every $\epsilon > 0$, a $(1-\epsilon)$ approximate solution can be computed in time $\widetilde{O}(k^2\epsilon^{-1}n^2)$. This follows from Proposition 22 and Lemma 23. This completes the proof Theorem 5. ◀

5.3 Linear-time algorithm for Uni-PBDS on Trees

We next establish our result for the scenario of Uni-PBDS on trees (Thm. 6). In fact, this result holds for a somewhat broader scenario, wherein, for each vertex x, the cardinality of $\text{PROB}_x = \{p_e \mid e \text{ is incident to } x\}$ is bounded above by some constant γ.

Proof of Theorem 6. Observe that the only place where approximation was used in our PTAS was in bounding the number of distinct values that can be taken by $A(\sigma)$ in Eq. (3). In order to obtain an exact solution for the bounded probabilities setting, the only modification performed in our algorithm is to redefine $A(\sigma)$ as follows.

$$A(\sigma) = \omega_v \cdot I_{(curr=1)} + \omega_v\Big(1 - (1 - par\cdot p_{(z_0,v)}) \cdot \prod_{\substack{i\in[1,t]\\ curr_i=1}} (1-p_{(z_i,v)})\Big)\cdot I_{(curr\neq 1)}.$$

It can be verified that the algorithm correctly computes $\mathcal{Y}_x$ at each step, that is, $\delta(x)$ is essentially zero. The time it takes to compute $\mathcal{Y}_v(par,curr,b)$, for a non-leaf v, crucially depends on the cardinality of $\{A(\sigma) \mid \sigma\in\mathcal{L}^*_t(b)\}$, where t is the number of children of v in $\mathcal{T}$. Observe that the number of distinct values $A(\sigma)$ can take is at most $b^{|\text{PROB}_x|} = O(k^\gamma)$. This along with Lemma 22 implies that the total runtime of our exact algorithm is $\widetilde{O}(k^{\gamma+2}n)$. ◀

5.4 Solving PBDS optimally on general trees

Let $c \geq 1$ be the smallest real such that 2-SPM problem has an $\widetilde{O}(N^c)$ time algorithm. We will show that in such a case, k-PBDS can be solved optimally on trees with arbitrary probabilities in $\widetilde{O}((\delta N)^{c\lceil k/2 \rceil + 1})$ time, for a constant $\delta > 0$.

For any node $v \in \mathcal{T}$, let $\mathcal{T}_v^i$, for $1 \leq i \leq \deg(v)$, represent the components of the subgraph $\mathcal{T} \setminus \{v\}$. We start with the following lemma (proved in Appendix B.1), which is easy to prove using a standard counting argument.

▶ **Lemma 24.** *For any set S of size k in $\mathcal{T}$, there exist a node $v \in \mathcal{T}$ and an index $q \in [1, \deg(v)]$ such that the cardinalities of the sets $S \cap \big(\bigcup_{i \lesssim q} T_v^i\big)$, $S \cap \big(T_v^q\big)$ and $S \cap \big(\bigcup_{i \gtrsim q} T_v^i\big)$ are all bounded by $k/2$.*

For the rest of this section, we refer to a tuple (v, q) satisfying the conditions stated in Lemma 24 as a *valid pair*. Let us suppose we are provided with a valid pair (v, q). For sake of convenience, we assume that $\mathcal{T}$ is rooted at node v. Let $U_0 = T_v^q$, $U_1 = \bigcup_{i \lesssim q} T_v^i$, and $U_2 = \bigcup_{i \gtrsim q} T_v^i$. Also let $\{x_1, \ldots, x_\alpha\}$ be the children of v in U_1, where $\alpha = q - 1$; let $\{y_1, \ldots, y_\beta\}$ be the children of v in U_2, where $\beta = \deg(v) - q$; and let z be q^{th} child of v.

An important observation is that if the optimal S contains v, then the problem is easily solvable in $O(k^2 \cdot n^{k/2})$ time, as then the structures U_0, U_1 and U_2 become independent. Indeed, it suffices to consider all $O(k^2)$ partitions of $k-1$ into triplet (c_0, c_1, c_2) consisting of integers in the range $[0, k/2]$, and next we iterate over all the $c_i \leq k/2$ subsets in U_i. This takes in total $O(k^2 n^{k/2})$ time. So the challenge is to solve the problem when v is not contained in S. Assuming (v, q) is a valid pair, and v is not contained in the optimal S, the solution S to k-PBDS is the union $S_0 \cup S_1 \cup S_2$ of the tuple $(S_0, S_1, S_2) \in U_0 \times U_1 \times U_2$ that maximizes

$$\mathcal{C}(U_0, S_0) + \mathcal{C}(U_1, S_1) + \mathcal{C}(U_2, S_2) + \omega_v \Big(1 - (1 - d \cdot p(v, z)) \cdot \prod_{\substack{i \in [1,\alpha] \\ x_i \in S_1}} (1 - p(v, x_i)) \prod_{\substack{j \in [1,\beta] \\ y_i \in S_2}} (1 - p(v, y_j))\Big) \tag{6}$$

where d is an indicator variable denoting whether or not $z \in S_0$, and $|S_1|, |S_2|, |S_3|$ are integers in the range $[0, k/2]$ that sum up to k.

Define Γ to be set of all quadruples (d, c_0, c_1, c_2) comprising of integers in the range $[0, k/2]$ such that $d \in \{0, 1\}$ and $c_0 + c_1 + c_2 = k$. For each $\gamma = (d, c_0, c_1, c_2) \in \Gamma$, let

$$\mathcal{L}_\gamma^1 = \left\{ \left(\frac{\mathcal{C}(U_1, S_1)}{\omega_v(1 - d \cdot p(v, z))}, \prod_{\substack{i \in [1,\alpha] \\ x_i \in S_1}} (1 - p(v, x_i)) \right) \;\middle|\; S_1 \subseteq U_1 \text{ is of size } c_1 \right\},$$

$$\mathcal{L}_\gamma^2 = \left\{ \left(\frac{\mathcal{C}(U_2, S_2)}{\omega_v(1 - d \cdot p(v, z))}, \prod_{\substack{j \in [1,\beta] \\ y_j \in S_2}} (1 - p(v, y_j)) \right) \;\middle|\; S_2 \subseteq U_2 \text{ is of size } c_2 \right\},$$

$$Z_\gamma = \max \left\{ \mathcal{C}(U_0, S_0) \;\middle|\; S_0 \subseteq U_0 \text{ is of size } c_0, \text{ and } d = I_{z \in S_0} \right\}.$$

So, the maximization considered in Eq. (6) is equivalent to the following optimization:

$$\max_{\substack{\gamma = (d, c_0, c_1, c_2) \in \Gamma \\ (a,b) \in \mathcal{L}_\gamma^1,\ (\bar{a}, \bar{b}) \in \mathcal{L}_\gamma^2}} \omega_v + Z_\gamma + \Big(\omega_v(1 - d \cdot p(v, z))\Big)\Big(a + \bar{a} - b\bar{b}\Big). \tag{7}$$

In the next lemma we show that optimizing the above expression is equivalent to solving $|\Gamma| = O(k^2)$ different 2-SPM problems (each of size $O(n^{k/2})$).

▶ **Lemma 25.** *Let* $A = \big((a_1, b_1), \ldots, (a_N, b_N)\big)$ *and* $\bar{A} = \big((\bar{a}_1, \bar{b}_1), \ldots, (\bar{a}_N, \bar{b}_N)\big)$ *be two arrays. Then, solving the maximization problem* $\max_{i_0,j_0}(a_{i_0} + \bar{a}_{j_0} - b_{i_0}\bar{b}_{j_0})$*, can be transformed in linear time to the following equivalent* 2-SPM*:*

$$\mathcal{L} = \big((Q + a_1, Rb_1), \ldots, (Q + a_N, Rb_N), (-Q + \bar{a}_1, R^{-1}\bar{b}_1), \ldots, (-Q + \bar{a}_N, R^{-1}\bar{b}_N)\big) \ ,$$

where $Q = \max_{i,j}(b_i\bar{b}_j) + 2\max_{i,j}(a_i + \bar{a}_j)$ *and* $R = \sqrt{4Q/\min_i(b_i)^2}$.

Proof. Consider the following 2-SPMs obtained by two equal partitions of $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}^1 &= \big((Q + a_1, Rb_1), \ldots, (Q + a_N, Rb_N)\big) \ , \text{ and} \\ \mathcal{L}^2 &= (-Q + \bar{a}_1, R^{-1}\bar{b}_1), \ldots, (-Q + \bar{a}_N, R^{-1}\bar{b}_N)\big) \ . \end{aligned}$$

Observe that the optimal value of $\mathcal{L}^1$ is bounded above by $2Q + 2\max_i(a_i) - R^2\min_i(\bar{b}_i)^2$ which is strictly less than $-Q$. Similarly, the optimal value of $\mathcal{L}^2$ is bounded above by $2Q + 2\max_i(\bar{a}_i) - \min_i(\bar{b}_i)^2/R^2$, which is again strictly less than $-Q$.

Now the answer to the optimization problem $\max_{i_0,j_0}(a_{i_0} + \bar{a}_{j_0} - b_{i_0}\bar{b}_{j_0})$ is at least $-Q$. This clearly shows that the solution to $\mathcal{L}$ cannot be obtained by its restrictions $\mathcal{L}^1$ and $\mathcal{L}^2$. Hence, the maximization problem $\max_{i_0,j_0}(a_{i_0} + \bar{a}_{j_0} - b_{i_0}\bar{b}_{j_0})$ is equivalent to solving the 2-SPM $\mathcal{L}$. ◀

Proof of Theorem 4. The time to compute $\mathcal{L}^1_\gamma$, $\mathcal{L}^2_\gamma$, Z_γ, for a given γ, is $\tilde{O}(n^{k/2})$. By transformation presented in Lemma 25, it follows that the total time required to optimize the expression in Eq. (7) is $k^{O(1)} \cdot n^{c\lceil k/2\rceil}$, which is at most $O\big((\delta n)^{c\lceil k/2\rceil+1}\big)$, for some constant $\delta \geq 1$. Now recall that Eq. (7) provides an optimal solution assuming (v, q) is a valid pair, and v is not contained in optimal S. Even when (v, q) is a valid pair, and v is contained in the optimal S, the time complexity turns out to be $O(k^2 \cdot n^{k/2})$. Iterating over all choices of pair (v, q) incurs an additional multiplicative factor of n in the runtime. ◀

6 Parameterization based on graph structure

In this section, we state our results on structural parameterizations of the Uni-PBDS problem. First, following the approach of Arnborg et al. [6], we formulate the MSOL formula for the Uni-PBDS problem where the quantifier rank of the formula is $O(k)$ (outlined in Appendix C.1). This indeed yields an FPT algorithm for the Uni-PBDS problem parameterized by budget k and the treewidth of the input graph.

In addition, we show that the Uni-PBDS problem is FPT for the budget parameter on apex-minor-free graphs. In particular we show that, for any integer k, and any n-vertex weighted apex-minor free graph with uniform edge probability, the Uni-PBDS problem can be solved in time $(2^{O(\sqrt{k}\log k)}n^{O(1)})$.

Due to space constraints, we defer our complete constructions and proofs to the upcoming full-version of the paper.

References

1 A. Abboud and K. Lewi. Exact weight subgraphs and the k-sum conjecture. In *Proc. 40th Int. Colloq. on Automata, Languages, and Programming (ICALP)*, pages 1–12. Springer, 2013.

2 A. Abboud, K. Lewi, and R. Williams. Losing weight by gaining edges. In *Proc. 22th European Symp. on Algorithms (ESA)*, pages 1–12. Springer, 2014.

3 J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004.

4 J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. *J. Algorithms*, 52(1):26–56, 2004. doi:10.1016/j.jalgor.2004.03.005.

5 T.M. Apostol. *Calculus*. Number v. 1 in Blaisdell book in pure and applied mathematics. Blaisdell Pub. Co., 1969.

6 S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12:308–340, 1991.

7 S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14 6:1170–5, 2004.

8 J. Añez, T. De La Barra, and B. Pérez. Dual graph representation of transport networks. *Transportation Research Part B: Methodological*, 30(3):209–216, 1996. doi:10.1016/0191-2615(95)00024-0.

9 M. O. Ball. Complexity of network reliability computations. *Networks*, 10(2):153–165, 1980. doi:10.1002/net.3230100206.

10 M. O. Ball and J. S. Provan. Calculating bounds on reachability and connectedness in stochastic networks. *Networks*, 13(2):253–278, 1983. doi:10.1002/net.3230130210.

11 F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In *Proc. 20th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 1316–1325, 2014.

12 C. J. Colbourn and G. Xue. A linear time algorithm for computing the most reliable source on a series-parallel graph with unreliable edges. *Theoretical Computer Science*, 209(1):331–345, 1998. doi:10.1016/S0304-3975(97)00124-2.

13 B. Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

14 B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*, volume 138 of *Encycl. Mathematics and Its Applications*. Cambridge Univ. Press, 2012.

15 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

16 M. S. Daskin. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, 1983. URL: http://EconPapers.repec.org/RePEc:inm:ortrsc:v:17:y:1983:i:1:p:48-70.

17 Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

18 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

19 W. Ding. Computing the most reliable source on stochastic ring networks. In *2009 WRI World Congress on Software Engineering*, volume 1, pages 345–347, May 2009. doi:10.1109/WCSE.2009.31.

20 W. Ding. Extended most reliable source on an unreliable general network. In *2011 International Conference on Internet Computing and Information Services*, pages 529–533, September 2011. doi:10.1109/ICICIS.2011.138.

21 W. Ding and G. Xue. A linear time algorithm for computing a most reliable source on a tree network with faulty nodes. *Theoretical Computer Science*, 412(3):225–232, 2011. Combinatorial Optimization and Applications. doi:10.1016/j.tcs.2009.08.003.

22 Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM. doi:10.1145/502512.502525.

23 Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In *Complexity Theory: Current Research, Dagstuhl Workshop, February 2-8, 1992*, pages 191–225, 1992.

24 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

25 P. G. Drange, M. S. Dregi, F. V. Fomin, S. Kreutzer, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, F. Reidl, F. S. Villaamil, S. Saurabh, S. Siebertz, and S. Sikdar. Kernelization and sparseness: the case of dominating set. In *Proc. 33rd Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 31:1–31:14, 2016.

26 T. Erlebach, M. Hoffmann, D. Krizanc, M. Mihalák, and R. Raman. Computing minimum spanning trees with uncertainty. In *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France, February 21-23, 2008, Proceedings*, pages 277–288, 2008. doi:10.4230/LIPIcs.STACS.2008.1358.

27 J. R. Evans. Maximum flow in probabilistic graphs-the discrete case. *Networks*, 6(2):161–183, 1976. doi:10.1002/net.3230060208.

28 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Subexponential algorithms for partial cover problems. *Inf. Process. Lett.*, 111(16):814–818, 2011. doi:10.1016/j.ipl.2011.05.016.

29 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 503–510. SIAM, 2010.

30 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Kernels for (connected) dominating set on graphs with excluded topological minors. *ACM Trans. Algorithms*, 14(1):6:1–6:31, 2018. doi:10.1145/3155298.

31 H. Frank and S. Hakimi. Probabilistic flows through a communication network. *IEEE Transactions on Circuit Theory*, 12(3):413–414, September 1965. doi:10.1109/TCT.1965.1082452.

32 M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposalbe graphs. In *Proc. 26th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 331–340, 1999.

33 Heng Guo and Mark Jerrum. A polynomial-time approximation algorithm for all-terminal network reliability. *SIAM J. Comput.*, 48(3):964–978, 2019. doi:10.1137/18M1201846.

34 R. Hassin, R. Ravi, and F. S. Salman. *Tractable Cases of Facility Location on a Network with a Linear Reliability Order of Links*, pages 275–276. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

35 R. Hassin, R. Ravi, and F. S. Salman. Multiple facility location on a network with linear reliability order of edges. *Journal of Combinatorial Optimization*, pages 1–25, 2017.

36 Dorit S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.

37 M. Hua and J. Pei. Probabilistic path queries in road networks: Traffic uncertainty aware path selection. In *Proc. 13th ACM Conf. on Extending Database Technology (EDBT)*, pages 347–358, 2010.

38 R. M. Karp and M. Luby. Monte-carlo algorithms for the planar multiterminal network reliability problem. *J. Complexity*, 1(1):45–64, 1985. doi:10.1016/0885-064X(85)90021-4.

39 D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. Ninth ACM Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 137–146, 2003.

40 Samir Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999. doi:10.1016/S0020-0190(99)00031-9.

41 T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.

42 J. Kneis, D. Mölle, and P. Rossmanith. Partial vs. complete domination: T-dominating set. In *Proc. 33rd Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 367–376. Springer-Verlag, 2007.

43 Emanuel Melachrinoudis and Mary E. Helander. A single facility location problem on a tree with unreliable edges. *Networks*, 27(4):219–237, 1996. doi:10.1002/(SICI)1097-0037(199605)27:3.

44 N. S. Narayanaswamy, M. Nasre, and R. Vijayaragunathan. Facility location on planar graphs with unreliable links. In *Proc. 13th Computer Science Symp in Russia (CSR)*, pages 269–281, 2018.

45 N. S. Narayanaswamy and R. Vijayaragunathan. Parameterized optimization in uncertain graphs - A survey and some results. *Algorithms*, 13(1):3, 2020.

46 G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978. doi:10.1007/BF01588971.

47 M. Patrascu. Towards polynomial lower bounds for dynamic problems. In *Proc. 42nd ACM Symp. on Theory of Computing (STOC)*, pages 603–610, 2010.

48 M. Patrascu and R. Williams. On the possibility of faster SAT algorithms. In *Proc. 21st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1065–1075, 2010.

49 Y. Peng, Y. Zhang, W. Zhang, X. Lin, and L. Qin. Efficient probabilistic k-core computation on uncertain graphs. In *Proc. 34th IEEE Conf. on Data Engineering (ICDE)*, pages 1192–1203, 2018.

50 Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms*, 9(1):11, 2012. doi:10.1145/2390176.2390187.

51 J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983. doi:10.1137/0212053.

52 G. Swamynathan, C. Wilson, B. Boe, K. C. Almeroth, and B. Y. Zhao. Do social networks improve e-commerce?: a study on social marketplaces. In *Proc. 1st Workshop on Online Social Networks (WOSN)*, pages 1–6, 2008.

53 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979. doi:10.1137/0208032.

54 Douglas R. White and Frank Harary. The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 31(1):305–359, 2001. doi:10.1111/0081-1750.00098.

55 Zhaonian Zou and Jianzhong Li. Structural-context similarities for uncertain graphs. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 1325–1330, 2013. doi:10.1109/ICDM.2013.22.

A Remainder of Section 4

A.1 Figures illustrated in the reduction

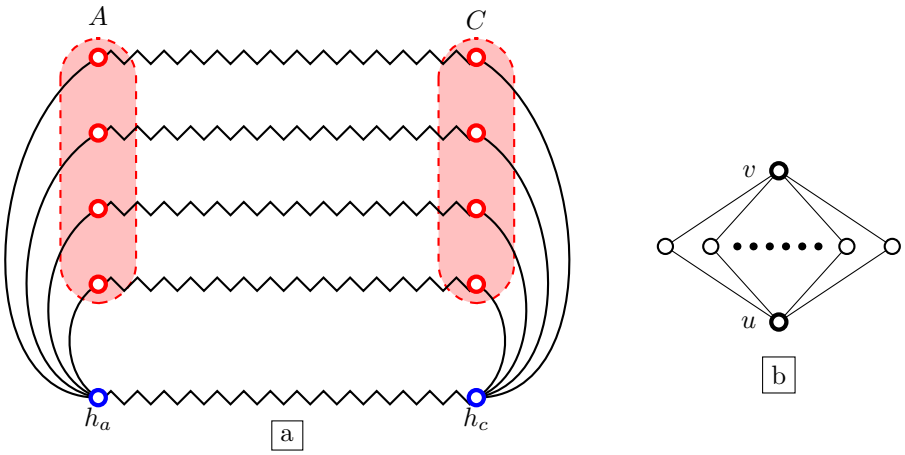

Figure 2 (a) The gadget $\mathcal{I}$ for $n = 4$. (b) The gadget $\mathcal{D}$. The zigzag edges in $\mathcal{I}$ between two vertices u and v is replaced by the gadget $\mathcal{D}_{u,v}$.

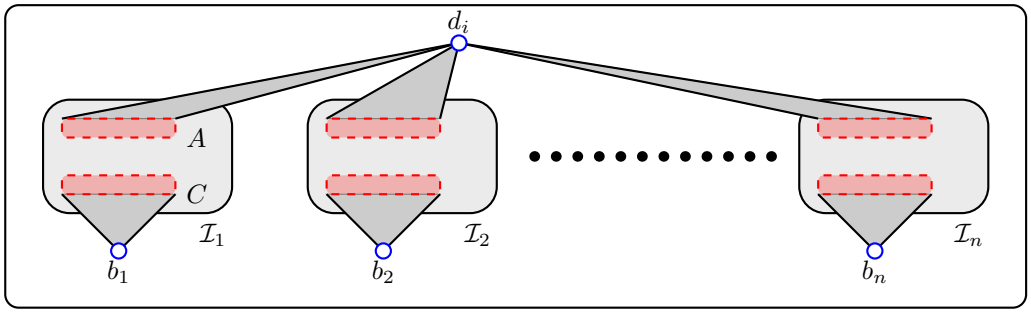

Figure 3 Illustration of a vertex block $\mathcal{G}_i$ for a V_i, $i \in [k]$. Note the n $\mathcal{I}$ gadgets for the n vertices in V_i. Similarly, an edge block $\mathcal{G}_{i,j}$ for some $1 \leq i < j \leq k$ has $|E_{i,j}|$-many $\mathcal{I}$ gadgets.

B Deferred Details from Section 5.4

B.1 Solving PBDS exactly on Trees

▶ **Reminder of Lemma 24.** For any set S of size k in $\mathcal{T}$, there exists a node $v \in \mathcal{T}$ and an index $q \in [1, \deg(v)]$ such that the cardinality of sets: $S \cap \left(\cup_{i \lneq q} T_v^i \right)$, $S \cap \left(T_v^q \right)$, and $S \cap \left(\cup_{i \gneq q} T_v^i \right)$, are all bounded by $k/2$.

Proof. We first show that there exists a node v in $\mathcal{T}$ satisfying the property $|S \cap T_v^i| \leq k/2$, for each $i \in [1, \deg(v)]$. Consider a node $u \in \mathcal{T}$. If u satisfies the above mentioned property then we are done. Otherwise, there exists an index $j \in [1, \deg(u)]$ for which $|S \cap T_u^j| \gneq k/2$. This implies the number of elements of S lying in $\{u\} \cup \left(T_u^1 \cup \cdots \cup T_u^{j-1}\right) \cup \left(T_u^{j+1} \cup \cdots \cup T_u^{\deg(u)}\right)$ is at most $k/2$. In such a case we replace u by its j^{th} neighbor. Repeating the process eventually leads to the required node v.

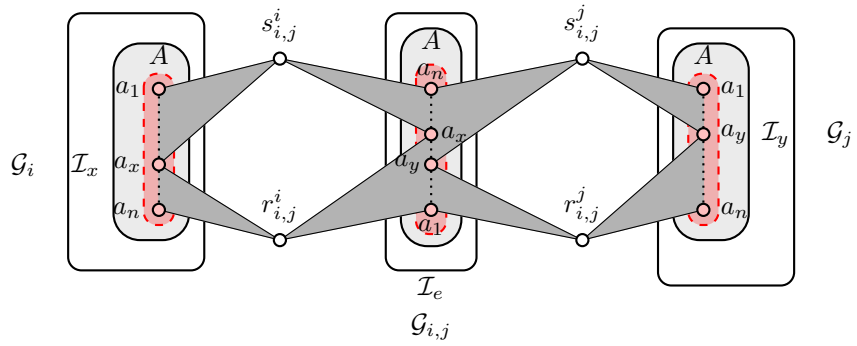

Figure 4 An illustration of the connector vertices $s^i_{i,j}$, $r^i_{i,j}$, $s^j_{i,j}$ and $r^j_{i,j}$, which connect the blocks $\mathcal{G}_i$ and $\mathcal{G}_{i,j}$, and $\mathcal{G}_j$ and $\mathcal{G}_{i,j}$, for some $1 \leq i < j \leq k$. The gadget $\mathcal{I}_e$ represents an edge $e = u_{i,x}u_{j,y} \in E_{i,j}$.

Now, let $q \in [1, \deg(v)]$ be the smallest integer for which $S \cap \left(T^1_v \cup \cdots \cup T^q_v\right)$ is larger than $k/2$. Then, $S \cap \left(\bigcup_{i \leq q} T^i_v\right)$ and $S \cap \left(\bigcup_{i \geq q} T^i_v\right)$ are both bounded by $k/2$, by definition of q. Also, $S \cap T^q_v$ is bounded by $k/2$ due to the choice of v. ◀

C Deferred details from Section 6

C.1 MSOL formulation of the Uni-PBDS problem

We show that an extension of Courcelle's theorem due to Arnborg et. al. [6] results in an FPT algorithm for the combined parameters treewidth and k. This is obtained by expressing the Uni-PBDS problem as a *monadic second order logic* (MSOL) formula (see [13, 14]) of length $O(k)$. The following MSOL formulas are used in the MSOL formula for the Uni-PBDS problem. The upper case variables (with subscripts) take values from the set of subsets of V, and the lower case variables take values from V.

- The vertex set S contains d elements.

$$\texttt{SIZE}_d(S) = \exists x_1 \exists x_2 \cdots \exists x_d \forall y (y \in S \rightarrow \bigvee_{i=0}^{d} (y = x_i))$$

- Given a vertex set X and a vertex x, there exists a set $S \subseteq X$ of size d, and for each vertex y in S, y is a neighbor of x.

$$\texttt{INC}_d(x, X) = \exists S(\texttt{SIZE}_d(S) \wedge \forall y((y \in S \rightarrow y \in X) \wedge (y \in S \rightarrow adj(x, y))))$$

- The sets $X, Y_0, Y_1, \ldots, Y_k$ partition the vertex set V.

$$\texttt{PART}(X, Y_0, Y_1, \ldots, Y_k) = \forall x \Big(\quad (x \in X \vee \bigvee_{i=0}^{k} x \in Y_i) \wedge \bigwedge_{i=0}^{k} \neg(x \in X \wedge x \in Y_i) \wedge \bigwedge_{i \neq j} \neg(x \in Y_i \wedge x \in Y_j) \Big)$$

Now we define the MSOL formula for the Uni-PBDS problem. The formula expresses the statement that V can be partitioned into X and $V \setminus X$, and $V \setminus X$ can be partitioned into $k+1$ sets $Y_0, Y_1, \ldots, Y_k$ such that for each set Y_i and each vertex y in Y_i, y has i neighbors in X.

$$\texttt{Uni-PBDS} = \exists X \exists Y_0 \exists Y_1 \cdots \exists Y_k \Big(\quad \texttt{PART}(X, Y_0, Y_1, \ldots, Y_k) \wedge$$
$$\forall x \forall y \Big(\big(y \in Y_0 \wedge x \in X \big) \rightarrow \neg \big(adj(x, y) \big) \Big)$$
$$\forall y \Big(\bigwedge_{i=1}^{k} \big(y \in Y_i \rightarrow \texttt{INC}_i(y, X) \big) \Big) \Big)$$

▶ **Lemma 26.** *The* quantifier rank *of* Uni-PBDS *is* $O(k)$.

Proof. There are $k + 2$ initial quantifiers for the sets $X, Y_0, Y_1, \ldots, Y_k$. For two MSOL formulas ϕ and ψ with quantifier rank $\text{qr}(\phi)$ and $\text{qr}(\psi)$, respectively, $\text{qr}(\phi \wedge \psi) = \text{qr}(\phi \vee \psi) = \max\{\text{qr}(\phi), \text{qr}(\psi)\}$. Therefore, $\text{qr}(\texttt{Uni-PBDS})$ is bounded as follows:

$$\begin{aligned} \text{qr}(\texttt{Uni-PBDS}) &= k + 2 + \max\{\text{qr}(\texttt{PART}), 1 + \text{qr}(\texttt{INC})\} \\ &= k + 2 + \max\{1, 1 + \max\{\text{qr}(\texttt{SIZE}), 2\}\} \\ &\leq k + 2 + k = 2k + 3 = O(k) \end{aligned}$$

◀

We now show that the Uni-PBDS problem is fixed-parameter tractable in parameters k and treewidth by expressing the maximization problem on the MSOL formula as a minor variation of *extended monadic second-order extremum problem* as described by Arnborg et. al. [6].

Proof of Theorem 8. For each $0 \leq i \leq k$, define the weight function w^i associated with the set variable Y_i as follows: for each $v \in V(G)$, $w^i_v = (1-(1-p)^i)w(v)$. The difference between the weight function in [6] and our problem is that in their paper $w(v)$ is considered to be constant value, for all vertices, for the set variable Y_i. Observe, however, that the running time of their algorithm does not change as long as w^i_v can be computed in polynomial time, which is the case in our definition. Therefore, our maximization problem is now formulated as a variant of the EMS maximization problem in [6]:

$$Maximize \sum_{u \in X} w(u) + \sum_{i=0}^{k} \sum_{u \in Y_i} w^i_v \cdot y^i_v \text{ over partitions } (X, Y_0, Y_1, \ldots, Y_k) \text{ satisfying } \texttt{Uni-PBDS}$$

Using Theorem 5.6 in [6] along with the additional observation, we make, that w^i_v can be efficiently computed, an optimal solution for the Uni-PBDS problem can be computed in time $f(\text{qr}(\texttt{Uni-PBDS}), w) \cdot poly(n)$, where $f(\text{qr}(\texttt{Uni-PBDS}), w)$ is a function which does not depend on n- it depends only on the quantifier rank of Uni-PBDS and the treewidth. By Lemma 26, $\text{qr}(\texttt{Uni-PBDS}) = O(k)$, and thus by [6], $f(\text{qr}(\texttt{Uni-PBDS}), w) = f(O(k), w)$. This shows that Uni-PBDS is FPT with resepect the parameters k and treewidth. Hence the theorem is proved. ◀