algorithms

Article

Parameterized Optimization in Uncertain Graphs—A Survey and Some Results

N. S. Narayanaswamy * and R. Vijayaragunathan *

Department of Computer Science and Engineering, Indian Institute of Technology Madras (IIT Madras), Chennai 600036, India

* Correspondence: swamy@cse.iitm.ac.in (N.S.N.); vijayr@cse.iitm.ac.in (R.V.)

Received: 11 November 2019; Accepted: 12 December 2019; Published: 19 December 2019

Abstract: We present a detailed survey of results and two new results on graphical models of uncertainty and associated optimization problems. We focus on two well-studied models, namely, the Random Failure (RF) model and the Linear Reliability Ordering (LRO) model. We present an FPT algorithm parameterized by the product of treewidth and max-degree for maximizing expected coverage in an uncertain graph under the RF model. We then consider the problem of finding the maximal core in a graph, which is known to be polynomial time solvable. We show that the PROBABILISTIC-CORE problem is polynomial time solvable in uncertain graphs under the LRO model. On the other hand, under the RF model, we show that the PROBABILISTIC-CORE problem is W[1]-hard for the parameter d, where d is the minimum degree of the core. We then design an FPT algorithm for the parameter treewidth.

Keywords: probabilistic graphs; uncertain graphs; influence maximization; cascade failure; linear reliable ordering; expected coverage; probabilistic core.

1. Introduction

Network data analytics has come to play a key role in many scientific fields. A large body of such real-world networks have an associated uncertainty and optimization problems are required to be solved taking into account the uncertainty. Some of the uncertainty are due to the data collection process, machine-learning methods employed at preprocessing, privacy-preserving reasons and due to unknown causes during the operation of the network. Throughout this work, we study the case where the uncertainty is associated with the availability or the nature of relationship between the vertices of the network. The vertices themselves are assumed to be always available, in other words, the vertices are assumed to be certain. The concepts can be naturally modified to model uncertainty by associating uncertainty with the vertices. Road networks [1,2] are a natural source of optimization problems where the uncertainty is due to the traffic. Indeed, in uncertain traffic networks [2], the travel-time on a road is inherently uncertain. One way of modeling this uncertainty is by modeling the travel-time as a random variable. Indeed, the random variable is quite complex since the probability that it takes a certain value is dependent on other parameters like the day of the week and time of the day. However, our focus is only on the fact that uncertainty is modeled by an appropriately defined random variable. The natural optimization problem on an uncertain traffic network is to compute the expected minimum-time s-t path. In biological networks [3], the *protein-protein interaction* (PPI) network is an uncertain network. In a PPI network, proteins are represented by vertices and interaction between proteins are represented by edges. The interaction between proteins are derived through noisy and error-prone experiments which cause uncertainty. Sometimes the interactions are predicted by the nature of proteins instead of experiments. In this example, the uncertainty can be of two types: it can be on the presence or the absence of a protein-protein interaction or on the strength of an interaction between two proteins. Similarly, social networks are another example of uncertain networks where the members of the network are known, and the uncertainty is on the link between two members in the network. The interaction between members of an uncertain network associated with a social network are obtained using link prediction and by evaluating peer influence [4]. In all these three examples of networks with uncertainty, the uncertainty on the edges can be modeled as random variables. A random variable can be used to indicate the presence or the absence of an edge. In this case, the random variable takes values from the set $\{0,1\}$ with each value having an associated probability. A random variable can also be used to model the strength of the interaction between two entities in the network, in which case for an edge, the corresponding random variable takes values from a set of values and each value has an associated probability.

In this work, we survey the different models of uncertainty and the associated optimization problems. We then present our results on optimization problems on uncertain networks when the networks have bounded treewidth. The fundamental motivation for this direction of study is that a typical optimization problem on an uncertain graph is an expectation computation over many graphs implicitly represented by the uncertain graph. A natural question is there are problems which have efficient algorithms on a graph in which the edges have no uncertainty but become hard on uncertain graphs. The typical optimization problems considered on uncertain graphs are shortest path [5], reliability [6], minimum spanning trees [7,8], maxflows [9,10], maximum coverage [11–13], influence maximization [14] and densest subgraph [15,16]. This article is structured partly as a survey and partly as an original research article. In Section 1.1, we formally present the concepts in uncertain graphs and then present subsequent details on uncertain graphs in Section 2. In Section 1.2, we survey the different algorithmic results on uncertain graphs and in Section 1.3 we outline our results.

1.1. Uncertain Graphs-Definition and Semantics

We consider the graphs with uncertain edges and certain vertices. An uncertain graph, denoted by $\mathcal{G}$, is a triple that consists of a vertex set V, an edge set E and a set of *outcomes* $\{\mathcal{A}_e \mid e \in E\}$. For each edge $e \in E$, the outcome $\mathcal{A}_e$ is a set of values and an associated probability distribution on $\mathcal{A}_e$. The outcome $\mathcal{A}_e$ is considered to be an interval or a finite set. For each $e \in E$, if the outcome $\mathcal{A}_e$ is an interval, the associated probability distribution is called a continuous distribution, and if it is a finite set, the associated probability distribution is called a discrete distribution. In either case, the natural distribution is the uniform distribution. For the case in which for each edge $e \in E$, $\mathcal{A}_e$ is a closed interval $[\mathcal{L}_e, \mathcal{U}_e]$, the probability that e gets specific value in $\mathcal{L}_e < \mathcal{U}_e$ is $\frac{1}{\mathcal{U}_e - \mathcal{L}_e}$. In case for each $e \in E$, $\mathcal{A}_e$ is a finite set under the uniform distribution, the probability that e gets a specific value from $\mathcal{A}_e$ is $\frac{1}{|\mathcal{A}_e|}$. Such uncertain graphs were the focus of the earliest results [5,7,9,10] and we survey these in Section 1.2. However, in general, the uncertainty could be modeled by any probability distribution A_e. Therefore, the uncertain graph is a succinct representation of the set of all edge weighted graphs such that for each edge e its weight is a value from $\mathcal{A}_e$. This set is an uncountable set when $\mathcal{A}_e$ is an interval and it is $\prod_{e \in E} |\mathcal{A}_e|$ in case each $e \in E$, A_e is a finite set. In this paper, we present our results for the case when for each edge e, the outcome $\mathcal{A}_e$ is the set of values $\{0,1\}$. Naturally, 0 models the absence of an edge and 1 models the presence of an edge. An uncertain graph under this condition is a succinct representation of the set of all edge subgraphs of $G = (V, E)$. In this case, the uncertain graph $\mathcal{G}$ is represented as a triple (V, E, p) where $p : E \rightarrow [0,1]$ is a function defined on E and $p(e)$ is said to be the survival probability of the edge e. The failure probability of an edge e is $1 - p(e)$. The set of all graphs represented by an uncertain graph is well-known as the *possible world semantics* (PWS) [17,18] of the uncertain graph. For each $E' \subseteq E$, $H = (V, E')$ is called a *possible world* of $\mathcal{G}$ and this is denoted by the notation $H \sqsubseteq \mathcal{G}$. For an uncertain graph $\mathcal{G} = (V, E, p)$, there are $2^{|E|}$ possible worlds. An uncertain graph and two of its possible worlds are illustrated in Figure 1.

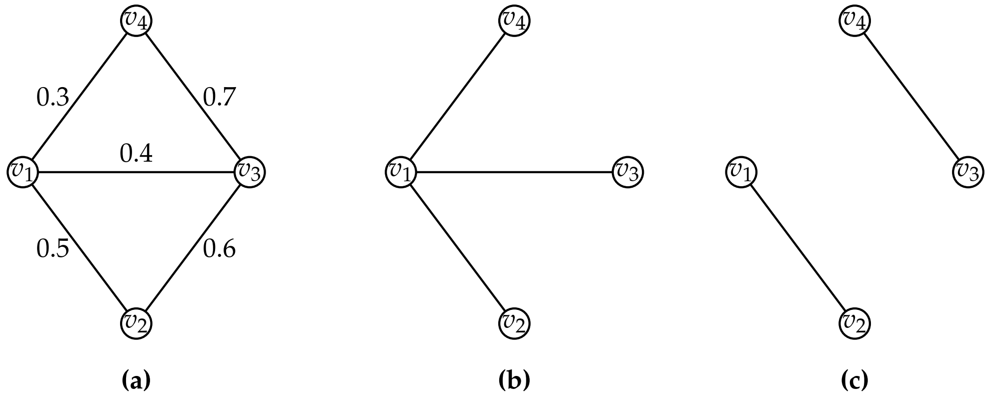

Figure 1. (**a**) A probabilistic graph $\mathcal{G} = (V, E, p)$; (**b**) A possible world H_1 of $\mathcal{G}$ with $P(H_1) = 0.0072$; (**c**) Another possible world H_2 of $\mathcal{G}$ with $P(H_2) = 0.0588$.

The probability associated with a possible world depends on the probability distributions on $\mathcal{A}_e$ and, most importantly, the dependence among the edge samples. A *distribution model* is a specification of the dependence among edge samples. Further, a distribution model uniquely determines a probability distribution on the possible worlds. For example, if the edge samples are all independent, the distribution model is called the Random Failure (RF) model. Under the RF model, the probability of a possible world H is given by $P(H) = \prod\limits_{e \in E(H)} p(e) \prod\limits_{e \in E \setminus E(H)} (1 - p(e))$. Based on the dependence among the edge samples, the literature is rich in different distributions on the possible worlds. The distributions of interest in this paper are the Random Failure (RF) model, Independent Cascade (IC) model, Set-based Dependency (SBD) model and the Linear Reliability Ordering (LRO) model. These distribution models are all well-motivated by practical considerations on uncertain graphs in influence maximization, facility location, network reliability, to name a few. The detailed description of these distribution models and the corresponding distributions on the possible worlds are discussed in Section 2. In short, an uncertain graph along with a distribution model is a succinct description of an unique probability distribution on the PWS of the uncertain graph. Therefore, an uncertain graph along with a distribution model is equivalent to a sampling procedure to obtain a random sample from the corresponding probability distribution on the PWS of the uncertain graph. In Section 2, we describe the different distribution models by describing the corresponding sampling procedures of the edges in E.

The typical computational problem is posed for a fixed nature of the dependence among the edge samples from an input uncertain graph. The distributions on the edges and the dependence among the edge samples uniquely define the probability distribution on the possible worlds. Therefore, for a fixed dependence among the edge samples, the generic computational problem is to compute a solution that optimizes the expected value of a function over the distribution on the possible worlds. The problems that have been extensively studied are facility location to maximize coverage [19] and a selection of influentials on a social network to maximize influence [14]. Clearly, both these problems are coverage problems, studied two decades apart motivated by different considerations. In this paper, we are also motivated by understanding the parameterized complexity of problems on uncertain graphs. Historically, the earliest results considered different graph problems in which the distributions are on the set of values of the edge weights. We next present a survey of those results.

1.2. Survey of Optimization Problems in Uncertain Graphs

Graphs have been used to represent the relationships between entities. A graph is denoted by $G = (V, E)$ where V is set of vertices and $E \subseteq V \times V$ is set of edges representing relationship between pairs of vertices. PPI networks in bioinformatics, road networks and social networks are graphs with an uncertainty among the edges and are considered as uncertain graphs. The earliest ideas in graphs with uncertain edge weights were introduced by Frank and Hakimi [9] in 1965. Frank and Hakimi studied the maximum flow in a directed graph $G = (V, E)$ with uncertain capacities. On an input, consisting of an uncertain digraph $G = (V, E)$ and a continuous random variable with the uniform probability distribution on the capacity of each edge e, the probabilistic flow problem is to find the maximum flow probability and the joint distribution of the cut set values. Later in 1969, under the same setting Frank [5] studied the probabilistic shortest path problem on an undirected graph $G = (V, E)$. The probabilistic shortest path problem is to compute for each ℓ, the probability that the shortest path is at most ℓ. In 1976, J. R. Evans [10] studied the probabilistic maximum flow in a directed acyclic graph (DAG) with a discrete probability distribution on the edge capacities. A relatively recent result in 2008 is on the minimum spanning tree problem on uncertain graphs, by Erlebach et al. [7], with uncertain edge weights from a continuous distribution. Here the goal is to optimize the number of edges whose weight is sampled to obtain a spanning tree which achieves the expected MST weight over the possible worlds. This problem is different from the optimization problems of interest to us in this paper- given an uncertain graph as input, our goal is to optimize the expected value of a structural parameter over the possible worlds of the uncertain graph.

Optimization problems on uncertain graphs related to connectedness are among the most fundamental problems. Apart from their practical significance, they pose significant algorithmic challenges in different computational models. Further, the computational complexity of the problems increases significantly in the presence of uncertainty. In 1979, Valiant [6] studied the network reliability problem on uncertain graphs. The network reliability problem is a well-studied #P-hard problem [6,17,20–23]. The reliability problems have numerous applications in communication networks [24,25], biological networks [3] and social networks [26,27]. For a given network, *reliability* is defined as the ability of the network to remain operational after the failure of some of its links. The input consists of an uncertain graph $\mathcal{G} = (V, E, p)$ and a subset $S \subseteq V$ and the aim is to compute the probability that S is connected. Clearly, the reliability problem on uncertain graphs generalizes the graph connectivity problem which is polynomial time solvable. The *Canadian Traveler Problem* (CTP), formulated by Papadimitriou and Yannakakis in 1991 [28], is an online problem on uncertain graphs. Given an uncertain graph $\mathcal{G} = (V, E, p)$, a source s and a destination t, a traveler must find a walk from s to t, where an edge e is known to have survived with probability $p(e)$ only when the walk reaches one of its end points, after which it does not fail, conditioned on it having survived. The objective is to minimize the expected path length over the distribution on the possible worlds and the walker's choices.

Coverage in Uncertain Graphs. Apart from the themes of network flows and connectedness, coverage problems are very practical when the uncertainty is on the survival of the edges. In this framework, the uncertainty is on whether an edge will survive a disaster and the goal is to place facilities in the network such that the expected coverage over the possible worlds is maximized. Each possible world can be thought of as the set of edges which survive a disaster. Formally, an uncertain graph $\mathcal{G} = (V, E, p)$ is a succinct description of the set of possible worlds that can arise due to a disaster in which an edge e is known to survive with probability $p(e)$. The nature of dependence among the edge samples is used to model the nature of edge failures in the event of a disaster. As mentioned earlier, a fixed nature of dependence among the edge samples uniquely defines a probability distribution on the possible worlds. From this point, in this paper, we consider uncertain graphs where the uncertainty is on the survival of an edge (recall, the other possibility is on the uncertain edge weight). Further, all the results we present are for a fixed dependence among edge samples and such a fixed dependence among the edge samples is called a *distribution model*. Given the motivation of disasters, the distribution model is specified based on the dependence among the edge failures (recall, the failure probability of an edge e is $1 - p(e)$). In this framework, for a fixed distribution model, the function to be optimized is the coverage function.

Definition 1 (Coverage within distance r)**.** *The input consists of an uncertain graph $\mathcal{G} = (V, E, p)$ and an integer k. The goal is to compute a k-sized vertex set S which maximizes the expected total weight of the vertices which are at distance at most r from S. The expectation is over the possible worlds represented by the uncertain graph.*

Naturally, we refer to k as the budget, r as the radius of coverage (the number of hops in the network) and S is the set of vertices where facilities have to be located. For certain graphs and $r = 1$, the facility location with unreliable edges problem is the well-studied *budgeted dominating set* problem which is known to be NP-hard [29]. The other case of natural interest is for $r = \infty$. In this case, the coverage problem is to find a set S of k vertices so as to maximize the expected number of vertices connected to S. In the case of certain graphs, this is polynomial time solvable as the problem is to find k connected components whose total vertex weight is the maximum.

Coverage in Social Networks. Coverage problems also have a natural interpretation in social networks. The dynamics of a social network based on the *word-of-mouth* effect have been of significant interest in marketing and consumer research [30], where a social network is referred to as an interpersonal network. The work by Brown and Reingen [30] state the different hypotheses for estimating the amount of uncertainty in a relationship between two persons in an interpersonal network. With the advent of social networks on the internet, the works of Domingos and Richardson [4,31] formalized the questions relating to the effective marketing of a product based on interpersonal relationships in a social network. In 1969, the work of Bass [32] had modeled the adoption of products as a diffusion process as a global phenomenon, independent of the interpersonal relationships between people in a society. Kempe, Kleinberg and Tardos (KKT) [14] brought together the earlier works on adoption of products in an interpersonal network and posed the question of selecting the most influential nodes in a social network with an aim to influence the maximum number of people to adopt a certain product or opinion. They considered the uncertainty in the social network to be the influence exerted by one individual on another individual and this is naturally modeled as an uncertain graph $\mathcal{G} = (V, E, p)$. The propagation of influence is modeled as a diffusion process which is a function of time as in Bass [32]. The influence maximization problems in KKT are considered under distribution models, which are described as diffusion phenomena, called the Independent Cascade model and the Linear Threshold Model. Among these two models, the Independent Cascade model is defined as a sampling procedure whose outcome is an edge subgraph of a given uncertain graph. Thus, this model is more relevant for our study of uncertain graphs and the IC model is defined in Section 2. In the influence maximization problems considered in KKT [14], the aim is to select k influential people S such that the expected number of people connected to S is maximized. The expected number of vertices, over the distribution model, connected to a set S is called the influence of a set S or the expected coverage of S and is denoted by $\sigma(S)$. Thus, the influence maximization problem is to find the set $\arg\max_{S \subseteq V, |S| = k} \sigma(S)$. In Section 2, we discuss the computational complexity of $\sigma(S)$ for different distribution models. Thus, the influence maximization problem in Reference [14] is essentially a facility location problem where the distribution model is specified by a diffusion phenomenon, $r = \infty$ and each input instance consists of an uncertain network and a budget k.

Coverage and Facility Location. The survey due to Snyder [33] in 2006 collects the vast body of results on the facility location problem in uncertain graphs into a single research article. The earliest result known to us, due to Daskin [19] who formulated the maximum expected coverage problem where vertices are uncertain, is different from our case where the uncertainty is on the edges and the vertices are known. To the best of our knowledge, it was Daskin's work [19] that considered the general setting of dependence among vertex failures. In this case, the probability distribution would have been defined uniquely on the possible worlds which would have been the set of induced subgraphs. Subsequently, many variants of the facility location problem for uncertain graphs have been studied for different distribution models where the uncertainty is on the survival of edges.

Eiselt et al. [34] considered the problem with single edge failure and for $r = \infty$. In this case, exactly one edge is assumed to have failed after a disaster and the objective is to place k facilities such that the expected weight of vertices not connected to any facility is minimized. In this case, each subgraph consisting of all the edges except one is a possible world. This problem is known to be polynomial time solvable for any $k \geq 1$. When the distribution model is the Random Failure model and $r = \infty$, the *most reliable source* (MRS) is well-studied [35–38]. The input is an uncertain graph and $k = 1$. The goal is to select one vertex v such that the expected number of vertices connected to v is maximized. Melachrinoudis and Helander [38] gave a polynomial time algorithm for the MRS problem on trees, followed by linear time algorithm on uncertain trees by Ding and Xue [37]. Colbourn and Xue [35] gave an $\mathcal{O}(n^2)$-time algorithm for the MRS problem on uncertain series-parallel graphs. Wei Ding [36] gave an $\mathcal{O}(n^2)$-time algorithm for the MRS problem on uncertain ring graphs.

Apart from the RF model, the Linear Reliability Ordering (LRO) distribution model has been well-studied recently. Under this distribution model, for each integer $r \geq 1$, the facility location problem is studied as MAX-EXP-COVER-r-LRO problem [11–13]. For the case when $r = \infty$, the problem is known as the MAX-EXP-COVER-LRO problem. Hassin et al. [11] presented an algorithm to solve the MAX-EXP-COVER-LRO problem via a reduction to the MAX-WEIGHTED-k-LEAF-INDUCED-SUBTREE problem. They then showed that the MAX-WEIGHTED-k-LEAF-INDUCED-SUBTREE problem can be solved in polynomial time on trees using a greedy algorithm. Consequently, they showed that the MAX-EXP-COVER-LRO problem can be solved in polynomial time. For $r = 1$, the MAX-EXP-COVER-1-LRO problem is NP-complete on planar graphs and the hardness follows from the hardness of budgeted dominating set due to Khuller et al. [29]. The MAX-EXP-COVER-1-LRO admits a $(1 - \frac{1}{e})$-approximation algorithm [12,13]. Similarly, Kempe et al. [14] shows that the influence maximization under the IC distribution model and the LT distribution model has a $(1 - \frac{1}{e})$ approximation algorithm. Both these results naturally follow due to the submodularity of the expected neighborhood size function, the monotonicity (as r increases) and submodularity of the expected coverage function and the result of Nemhauser et al. [39] on greedy maximization of submodular functions. In the setting of parameterized algorithms, the MAX-EXP-COVER-1-LRO problem is W[1]-complete for solution size as the parameter and this follows from the hardness of budgeted dominating set problem. An FPT algorithm for the MAX-EXP-COVER-1-LRO problem with treewidth as the parameter is presented in Reference [13]. Formally stated, given an instance $\langle \mathcal{G} = (V, E, p), k \rangle$ of the MAX-EXP-COVER-1-LRO problem, an optimal solution can be computed in time $4^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$ where t is the treewidth of the graph $G = (V, E)$ which is presented in the input as a tree decomposition.

Finding Communities in Uncertain Graphs. Finding communities is a significant problem in social network and in bioinformatics. A natural graph theoretic model for a community is a dense subgraph. A well-known dense subgraph is the *core*. Given an integer d, a graph $G = (V, E)$ is said to be d-core if degree of every vertex $v \in V$ is at least d. The way of obtaining the unique maximal induced subgraph of a graph G which is a d-core is to repeatedly discard vertices of degree less than d. If the procedure terminates with a non-empty graph, then the graph is a d-core of the graph G. A vertex v is said to be in a d-core if there is a d-core which contains v. This idea is generalized to the uncertain graph framework as follows: Given an uncertain graph $\mathcal{G} = (V, E, p)$ and the distribution model is the RF model, the *d-core probability* of a vertex $v \in V$, denoted by $q_d(v)$, is the probability that v is in the d-core of a possible world. In other words, $q_d(v) = \sum\limits_{H \sqsubseteq \mathcal{G}} P(H) I(H, d, v)$, where $I(H, d, v)$ is an indicator function that takes value one if and only if there is a d-core of H that contains v, and $P(H)$ is probability of the possible world H. In this setting, we consider the (d, θ)-core problem defined by Peng et al. [16]. We refer to this problem as the INDIVIDUAL CORE problem and it is defined as follows.

Definition 2 (INDIVIDUAL-CORE). *The input consists of an uncertain graph $\mathcal{G} = (V, E, p)$, an integer d and a probability threshold $\theta \in [0, 1]$. The objective is to compute an $S \subseteq V$ such that for each $v \in S$, $q_d(v) \geq \theta$.*

Peng et al. [16] shows that the INDIVIDUAL-CORE problem is NP-complete. Prior to the results of Peng et al. [16], Bonchi et al. [15] introduced the study of d-core problem in uncertain graphs. We refer to the d-core problem on uncertain graphs as the PROBABILISTIC-CORE problem defined the follows.

Definition 3 (PROBABILISTIC-CORE)**.** *Given an uncertain graph $\mathcal{G} = (V, E, p)$, an integer d and a probability threshold $\theta \in (0, 1]$, then the aim of the* PROBABILISTIC-CORE *problem is to find a set $K \subseteq V$ such that the* $\Pr(K$ *is a d-core in* $\mathcal{G})$ *is at least θ.*

The problem of deciding on the existence of such a set K can be shown to be NP-hard using the hardness result given in Reference [16]. On the other hand, if $p(e) = 1$ for all $e \in E$, then the d-problem is polynomial time solvable as described at the beginning of this discussion.

A chronological listing of different optimization problems on uncertain graphs is presented in Table 1. The tabulation shows that there are many distribution models under which different optimization problems could be considered. The goal would be to understand the complexity of computing that expectation when the input is presented as an uncertain graph. Indeed, any NP-Complete problem on certain graphs remains NP-Complete for any distribution model when the inputs are uncertain graphs. Therefore, our natural focus is on Exact and Parameterized Computation of the expectation on uncertain graphs. The goal of the area of exact and parameterized computation is to classify problems based on their computational complexity as a function of input parameters other than the input size. The desired solution size, the treewidth of an input graph, and the input size are natural well-studied parameters. An algorithm with running time $f(k)n^{\mathcal{O}(1)}$ is said to be a Fixed Parameter Tractable algorithm with respect to the parameter k. Interestingly, there are many problems that do not have FPT algorithms with respect to some parameters, while they have FPT algorithms with respect to others. A rich complexity theory has evolved over nearly four decades of research with the W-hierarchy being the central classification. In this hierarchy problem classes are ordered in increasing order of computational complexity. In this hierarchy the problems which have FPT algorithms are considered the *simplest* in terms of computational complexity. The complete history of this line of research can be found in the most recent textbook [40].

Table 1. A chronology of studies on uncertain graphs

Work	**Optimization Problem**	**Uncertainty Model**
Frank and Hakimi, 1965 [9]	Probabilistic maximum flow	Capacities on the edges are drawn from an independent continuous distribution.
Frank, 1969 [5]	Probabilistic shortest path	Length of the edges are drawn from a continuous distribution.
Evans, 1976 [10]	Probabilistic maximum flow	Capacities on the edges are obtained from an arbitrary but unknown discrete probability distribution.
Valiant, 1979 [6]	Network reliability	The probability $p(e)$ is the same for each edge and failure of every edge is independent.
Sigal, Pritsker and Solberg, 1980 [41]	Stochastic shortest path	Edge weights are drawn from a known cumulative distribution function.
Daskin, 1983 [19]	Expected coverage	Failure probability is the same for each vertex
Papadimitriou Yannakakis, 1991 [28]	Canadian Traveler Problem	Each edge has a survival probability, edge failure is independent and algorithm knows of the failure during execution.
Guerin and Orda, 1999 [42]	Most reliable path and flows with bandwidth selection	Each edge e has a survival probability $p_e(x)$ for the availability of bandwidth x.
Hassin, Salman and Ravi, 2009 (2017) [11,12]	Expected coverage	Each edge e has a survival probability $p(e)$ and edge failure follows LRO model.
Bonchi, Gullo, Kaltenbrunner and Volkovich, 2014 [15]	PROBABILISTIC-CORE	Each edge e has a survival probability $p(e)$ and edge failure follows RF model.

1.3. Our Questions and Results

Given our focus on exact and parameterized algorithms, we think that it should be possible to classify problems based on the hardness of efficiently computing the expectation over different distribution models. One of the contributions of this paper is to collect many of the known algorithmic results on uncertain graphs and the different distribution models for which results have been obtained. Our focus is on uncertain graph optimization problems on the LRO and RF distribution models, and these models have been the focus of many previous results in the literature. Our results add to the knowledge about these two models and are among the first parameterized algorithms on uncertain graphs. They also give an increased understanding of how treewidth of uncertain graphs can be used along with the structure of the distribution models to compute the expectation in time parameterized by the treewidth. Finally, the motivation for the choice of these two models is that the number of possible worlds with non-zero probability under the LRO model is equal to the number of edges m, while the number of possible worlds under the RF model is an exponential in m, which is the maximum number of possible worlds. We define the different distribution models listed in Section 1.1 in detail and some of their properties are identified from relationships between the models in Section 2.

Our first case study on parameterized algorithms under the RF model is on maximizing expected coverage in uncertain graphs. The starting point of our work is the reduction of the MAX-EXP-COVER-LRO problem to the MAX-WEIGHTED-k-LEAF-INDUCED-SUBTREE problem, which can be solved in polynomial time [11,12]. The reduction was from maximizing the expectation coverage to maximizing the total weight of a combinatorial parameter. However, the reduction does not work for the MAX-EXP-COVER-1-LRO, as the problem is at least as hard as budgeted dominating set. In the case when the graph has bounded treewidth, we were able to show in a previous work [13] that MAX-EXP-COVER-1-LRO has an FPT algorithm parameterized by treewidth. The dynamic programming algorithm depends on properties specific to the LRO distribution model. On the other hand, the status of the question is unclear when the distribution model is the RF model. We address this by presenting a DP algorithm for the MAX-EXP-COVER-1-RF problem in Section 4, which is an FPT algorithm parameterized by the product of the treewidth and max-degree of the input graph.

The second case study on parameterized algorithms under the RF model is on finding a d-core in uncertain graphs. In Section 5, we consider the PROBABILISTIC-CORE problem, and design a polynomial time algorithm for the LRO distribution model. For the RF distribution model, we observe that the PROBABILISTIC-CORE problem is W[1]-hard for the parameter d, where d is the minimum degree of the core. Then we design a DP algorithm, which is an FPT algorithm for PROBABILISTIC-CORE with respect to treewidth as the parameter.

Essentially in both the case studies, given an uncertain graph under the RF model, the DP uses the tree decomposition to efficiently compute the expected value, which is expressed as a weighted summation over the exponentially many possible worlds.

2. Distribution Models for Uncertain Graphs

As mentioned in Section 1.1, a distribution model along with an uncertain graph $\mathcal{G} = (V, E, p)$ uniquely describes the probability distribution on the possible worlds. We describe the distribution model by describing a corresponding sampling procedure on the edges of $\mathcal{G} = (V, E, p)$. This formalism standardizes the nomenclature of optimization problems on uncertain graphs. For example, for the coverage problems on uncertain graphs, the problem names MAX-EXP-COVER-1-LRO and MAX-EXP-COVER-1-RF clearly state the distribution model and the radius of coverage relevant for the problem. An instance of each of these problems is an uncertain graph $\mathcal{G} = (V, E, p)$ and an integer $k \geq 1$. In this section, we present an edge sampling procedure from a given uncertain graph $\mathcal{G} = (V, E, p)$ corresponding to a distribution model on $\mathcal{G}$. The outcome of a sampling procedure is a possible world, which is an edge subgraph H of the graph $G = (V, E)$. An edge $e \in E(H)$ is called a *survived edge* and an edge $e \in E \setminus E(H)$ is called a *failed edge*. We also present some new observations regarding the different distribution models.

2.1. Random Failure Model

Random Failure (RF) model is the most natural distribution model. In this sampling procedure, an edge $e \in E$ is selected with probability $p(e)$ independent of the outcome of every other edge in E. Thus, each edge subgraph of G is a possible world and for an edge subgraph H, the probability that H is the outcome of the sampling procedure is denoted by $P(H)$ which is given by the equation $P(H) = \prod_{e \in E_H} p(e) \prod_{e \in E \setminus E_H}(1 - p(e))$.

2.2. Independent Cascade Model

The sampling procedure to define the Independent Cascade (IC) model is very naturally described by a diffusion process which proceeds in rounds. The process is as defined in Reference [14]. The diffusion process starts after round 0. In round 0, A_0 is a non-empty set of vertices and these are called *active* vertices. The process maintains a set of edges A_t after round $t \geq 0$. The vertices in A_t are said to be active. The sampling process is as follows. If a vertex v is in A_{t-1}, then it remains active in round t also, that is $v \in A_t$. The diffusion process in round $t+1$ is as follows: If v is in $A_t \setminus A_{t-1}$, that is v first becomes active after round $t-1$, then consider each edge $e = uv$ such that u is not in A_t. Each such edge e is sampled with probability $p(e)$ independent of the other outcomes. If $e = uv$ is selected (that is, it survives), then u is added to A_{t+1}. We reiterate that the edges incident on v will not be sampled in subsequent rounds. The sampling procedure terminates in not more than $|E|$ rounds. The outcome of this sampling procedure is the edge subgraph formed by a set of those edges which were selected when the first end point of the edge becomes active. Kempe et al. [14] showed that for any edge subgraph H of $G = (V, E)$, the probability that H is the outcome is given by $P(H) = \prod_{e \in E(H)} p(e) \prod_{e \in E \setminus E(H)}(1 - p(e))$.

Observation 1. *For an uncertain graph $\mathcal{G} = (V, E, p)$ the RF Model and the IC Model are identical.*

The next distribution model is a generalization of the RF model and was introduced by Gunnec and Salman [43].

2.3. Set-Based Dependency (SBD) Model

The uncertain graph $\mathcal{G} = (V, E, p)$ satisfies the additional properties that E is partitioned into $\{E_1, E_2, \ldots E_t\}$ of E, for some $t \geq 1$. Further, p satisfies the property that for any two edges e_1 and e_2 that belong to the same part in the partition, $p(e_1) \neq p(e_2)$. Typically, this partition is a fixed partition of E coming from the domain where the edge failures occur according to the SBD model. The sampling procedure definition of the SBD model is as follows: the edge sets are considered in order from E_1 to E_t, and for $1 \leq i < j \leq t$, the edges in the set E_i are considered before the edges in the set E_j. For each $1 \leq i \leq t$, the edges in E_i are considered in decreasing order of the value p. For each $1 \leq i \leq t$, when an edge $e \in E_i$ is considered, it is sampled with probability $p(e)$ independent of the outcome of the previous samples. If the outcome selects the edge e (that is, if e survives), then the next edge in E_i is considered. Otherwise, all the remaining edges of E_i that are to be considered after e, are considered to have failed, and the set E_{i+1} is considered. The procedure terminates after considering $E_1, \ldots, E_t$. The edge subgraph consisting of the selected edges (survived edges) is the outcome of this sampling procedure. The number of possible worlds that have a non-zero probability of being an outcome of the sampling procedure is $\prod_{i=1}^{t}(|E_i| + 1)$. To summarize, under the SBD model, for each $1 \leq i \leq t$, the survival of an edge e in a set E_i implies that every edge $e' \in E_i$ with a greater survival probability than that of e would have survived. Further, the edge samples of edges in two different sets are mutually independent. As a consequence of this, we have the following observation.

Observation 2. *The RF model on an uncertain graph $\mathcal{G} = (V, E, p)$ is identical to the SBD model on $\mathcal{G}$ and E is partitioned into m sets each containing an edge of E.*

2.4. Linear Reliable Ordering (LRO) Model

The case when the partition of E consists of only one set, the SBD model is called the Linear Reliability Ordering (LRO) model introduced by Hassin et al. [11]. Let $p(e_m) < p(e_{m-1}) < \ldots < p(e_2) < p(e_1)$. Then, under the LRO model it follows that for each $j > i$, $\Pr(e_j \text{ fails } \mid e_i \text{ fails }) = 1$. Further, the possible worlds is the set of $m+1$ graphs $\{G_0, \ldots, G_m\}$ where $E(G_0) = \emptyset$, $E(G_m) = E$, and for each $0 \le i \le m-1$, $E(G_{i+1}) = E(G_i) \cup \{e_{i+1}\}$. The following lemma shown by Hassin et al. [11] regarding the probability distribution on the possible worlds uniquely defined by the LRO model.

Lemma 1 (Hassin et al.[11])**.** *For $0 \le i \le m$, the probability of the possible world G_i is given by*

$$P(G_i) = \begin{cases} 1 - p(e_1) & \textit{if } i = 0 \\ p(e_i) - p(e_{i+1}) & \textit{if } 1 \le i < m \\ p(e_m) & \textit{if } i = m. \end{cases}$$

Coming back to the SBD model, on the uncertain graph $\mathcal{G} = (V, E, p)$ and E partitioned into $\{E_1, E_2, \ldots E_\ell\}$ for some $\ell \ge 1$, the probability of an outcome H under the SBD model naturally follows from Lemma 1. The idea is to consider an outcome of the SBD model as ℓ independent samples from the LRO model on the uncertain graphs $(V, E_1, p_1), (V, E_2, p_2), \ldots, (V, E_\ell, p_\ell)$, where for each $1 \le i \le \ell$, p_i is the function p restricted to E_i.

This completes our description of the distribution models known in the literature. We next present two dynamic programming algorithms on the input uncertain graph $\mathcal{G} = (V, E, p)$, when the graph $G = (V, E)$ is presented as a nice tree decomposition. The algorithms compute an expectation under the RF model and have worse running times than that of corresponding algorithms for computing the expectation under the LRO model.

3. Definitions Related to Graphs

Every graph we consider in this work are simple undirected graphs unless. A graph $G = (V, E)$ is an undirected graph with vertex set V and edge set E. We denote the number of vertices and edges by n and m, respectively. 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. For each vertex $v \in V$, $deg_G(v)$ denote the degree of the vertex v in G. When G is clear in the context $deg(v)$ is used. The maximum degree of the graph G, denoted by $\Delta(G)$, and the minimum degree of the graph G, denoted by $\delta(G)$, are the maximum and minimum degree of its vertices. When G is clear in the context Δ and δ is used. Other than this, we follow the standard graph theoretic terminologies from Reference [44]. We define some special notations for the uncertain graphs as follows. Given a vertex $v \in V$, let $E(v) \subseteq E$ denote the set of all edges incident on v. Given a subset of vertices $C \subseteq V$, let $E(C) = \cup_{v \in C} E(v)$ denote the edge set of the vertex-induced uncertain subgraph $\mathcal{G}[C]$. Similarly, given an edge set $F \subseteq E$, let $V(F) = \cup_{e=uv \in F} \{u, v\}$ denote the vertex set of the edge-induced uncertain graph $\mathcal{G}[F]$.

We study the parameterized complexity of the coverage problems and k-core problem on uncertain graphs. We follow the standard parameterized complexity terminologies from Reference [40]. We define the parameter treewidth that will be relevant to our discussion.

Definition 4 (Tree Decomposition [45,46])**.** *A tree decomposition of a graph G is a pair (X, H) such that H is a tree and $X = \{X_i \subseteq V : i \in H\}$. For each node $i \in H$, X_i is referred to as* bag *of i. The following three conditions hold for a tree decomposition (X, H) of the graph G.*

(a) For each vertex $v \in V$, there is a node $i \in H$ such that $v \in X_i$.
(b) For each edge $uv \in E$, there is a node $i \in H$ such that $u, v \in X_i$.
(c) For each vertex $v \in V$, the induced subtree of the nodes in H that contains v is connected.

The *width* of a tree decomposition is the $\max_{i \in H}(|X_i| - 1)$. The *treewidth* of a graph, denoted by tw, is the minimum width over all possible tree decompositions of G. An example of a tree decomposition is illustrated in Figure 2. For our algorithm, we require a special kind of decomposition, called the *nice tree decomposition* which we define below.

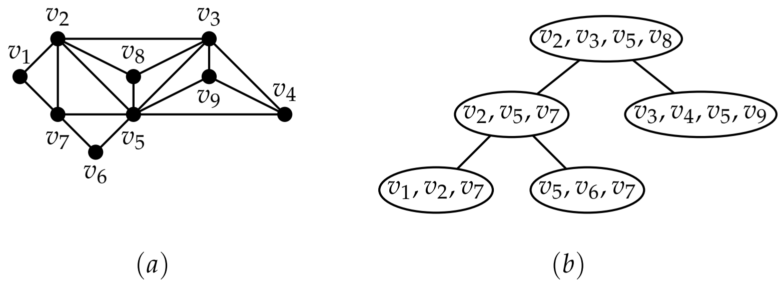

Figure 2. (**a**) A graph with 9 vertices; (**b**) An optimal tree decomposition with treewidth 3.

Definition 5 (Nice tree decomposition [46])**.** *A* nice tree decomposition *is a tree decomposition, rooted by a node r with $X_r = \emptyset$ and each node in the tree decomposition is one of the following four type of nodes.*

1. **Leaf node.** *A node $i \in H$ with no child and $X_i = \emptyset$.*
2. **Introduce node.** *A node $i \in H$ with one child j such that $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.*
3. **Forget node.** *A node $i \in H$ with one child j such that $X_i = X_j \setminus \{v\}$ for some $v \in X_j$.*
4. **Join node.** *A node $i \in H$ with two children j and g such that $X_i = X_j = X_g$.*

An example of four types of nodes is illustrated in Figure 3. Given a tree decomposition (X, H) of a graph G with width k, a nice tree decomposition (X', H') with same width and $\mathcal{O}(nk)$ nodes can be computed in time $\mathcal{O}(nk^2)$ [46]. Hereafter, we will assume that the tree decompositions considered are nice. For each node $i \in H$, let H_i denote the subtree rooted at i. Let $n_i = |X_i|$. Let X_i^+ be the set of all vertices in the bag of nodes in the subtree H_i.

$$X_i^+ = \begin{cases} X_i & \text{if } i \text{ is a leaf node} \\ X_i \cup \bigcup\limits_{j \in ch(i)} X_j^+ & \text{if } i \text{ is a non-leaf node,} \end{cases}$$

where $ch(i)$ is the set of all children of i in H. We use j and k to denote the two children of a join node i in H. Further, j denotes the left child, and g denotes the right child. In case i has only one child, as in the case of introduce and forget nodes, only the left child j is well-defined and g does not exist. In this case, X_g^+ is taken to be the empty set. We refer to Reference [40] for a thorough introduction to treewidth and its algorithmic properties.

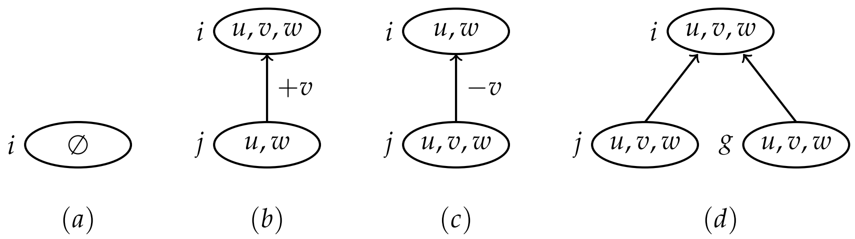

Figure 3. An example of leaf (**a**), introduce (**b**), forget (**c**) and join (**d**) nodes. Directed edges denote child to parent link.

4. Max-Exp-Cover-1-RF Problem is FPT by (Treewidth · Δ)

An instance of Max-Exp-Cover-1-RF consists of a tuple $\langle \mathcal{G}, w, k \rangle$ where $\mathcal{G} = (V, E, p)$ is an uncertain graph, w is a function that assigns weight $w(v)$ to each vertex v, and k is the budget. The goal is to find a set $S \subseteq V$ such that $|S| \leq k$ and the expected total weight of the vertices dominated by S is maximized. The expectation is computed over the probability distribution uniquely defined on the possible worlds by the RF model. We introduce the function $\mathcal{C}$ which we refer to as the *coverage* function. The two arguments of the coverage function are subsets T and S of the vertex set V, and the value $\mathcal{C}(T,S)$ is the expected coverage of T by S, where the expectation is computed over the possible worlds. For any subsets $S, T \subseteq V$, the *Coverage* of T by S, denoted by $\mathcal{C}(T,S)$, is $\sum_{v \in T} \mathcal{C}(v,S)$. For a vertex $v \in V$ and $S \subseteq V$, the expected coverage of v by the set S, denoted by $\mathcal{C}(v,S)$, is given by

$$\mathcal{C}(v,S) = \begin{cases} w(v) & \text{if } v \in S \\ w(v)\Big(1 - \prod_{u \in S \cap N(v)} (1 - p_{uv})\Big) & \text{otherwise.} \end{cases} \tag{1}$$

Note that if a set is a singleton set, then we abuse notation a little and write the element instead of the set.

Our algorithm for the MAX-EXP-COVER-1-RF follows the classical bottom-up approach for dynamic programming based on the nice tree decomposition. We compute a certain number of candidate solutions for the subproblem at each node $i \in H$, using only the candidate solution values maintained in the children of i. At each node $i \in H$, we have $(5 \cdot 2^{\Delta})^{n_i}$ candidate solutions which are stored in a table denoted by T_i. The final optimal solution is obtained from the solutions stored in T_r at the root node r of H. At any node i in H, the expected coverage of a vertex $u \in X_i$ by a set S in the subproblem at node i in H is computed by decomposing $N(u) \cap S$ carefully to avoid over-counting. Towards this end, we introduce notation for the coverage conditioned on the event that some edges have failed. We denote this event by a function f and call this the *surviving neighbors* (SN) function. The SN-function $f : V \to 2^V$ has the property that for each $u \in V$, $f(u) \subseteq N(u)$. In other words, we are interested in the expected coverage conditioned on the event that for each $u \in V$, all the edges to vertices in $N(u) \setminus f(u)$ have failed.

Definition 6. *Let u be a vertex and $S \subseteq V$ be a set and f be an SN-function such that $S \cap f(u) = S \cap N(u)$. The conditional coverage of u by S restricted by f is defined to be*

$$\mathcal{C}_f(u,S) = \Big(\prod_{v \in N(u) \setminus f(u)} (1 - p(uv))\Big)\mathcal{C}(u, S \cap f(u))$$

Extending this definition to sets, for any two vertex sets T, S, we define $\mathcal{C}_f(T,S) = \sum_{u \in T} \mathcal{C}_f(u,S)$.

For each $u \in V$, $\mathcal{C}_f(u,S)$ is the expected coverage of u by $S \cap f(u)$ conditioned on the event that u is not covered by any vertex from the set $N(u) \setminus f(u)$. Further, $\mathcal{C}_f(u,S)$ is the product of $w(u)$ and the probability of sampling a subgraph $G' \subseteq G$ in which u has a neighbor in $S \cap N(u)$ and no neighbor from $N(u) \setminus f(u)$. We refer to $\mathcal{C}_f(u,S)$ as the conditional coverage of u by S (we leave out the phrase *restricted by f*).

Lemma 2. *Let $u \in V$ and $X, Y \subseteq V$ such that $X \cap Y = \emptyset$, and consider an SN-function f such that $f(u) = N(u) \setminus X$. Then $\mathcal{C}(u, X \cup Y) = \mathcal{C}(u,X) + \mathcal{C}_f(u,Y)$.*

Proof.

$$\begin{aligned}
\mathcal{C}(u, X\cup Y) &= w(u)\Big(1-\prod_{v\in(X\cup Y)\cap N(u)}(1-p(uv))\Big)\\
&= w(u)\Big(1-\prod_{v\in X\cap N(u)}(1-p(uv))\cdot\prod_{v\in Y\cap N(u)}(1-p(uv))\Big)\\
&= w(u)\Big(\Big(1-\prod_{v\in X\cap N(u)}(1-p(uv))\Big)+\prod_{v\in X\cap N(u)}(1-p(uv))\Big(1-\prod_{v\in Y\cap N(u)}(1-p(uv))\Big)\\
&= \mathcal{C}(u,X)+\prod_{v\in X\cap N(u)}(1-p(uv))\mathcal{C}(u,Y)\\
&= \mathcal{C}(u,X)+\mathcal{C}_f(u,Y)
\end{aligned}$$

Hence the proof. □

4.1. Recursive Formulation of the Value of a Solution

Let $S\subseteq V$ be a set of size k for the MAX-EXP-COVER-1-RF problem in $\mathcal{G}$. We now present a recursive formulation to compute the expected coverage of V by S. Throughout this section S denotes this set of size k. Let i be a node in H. Let j and g be the children of i. In case i is either introduce node or forget node, then g is considered to be a null node with $X_g = X_g^+ = \emptyset$. In node i, let $A = S\cap X_i$, $S_i = S\cap X_i^+$ and $Z_i = S_i\setminus A$. Let $\hat{Z} = S\setminus S_i$. For the solution S, the set X_i can be partitioned into five sets (A, C, L, R, B) as follows.

$A = \{u\in X_i\cap S\}$

$C = \{u\in X_i\setminus A \mid u\in N(A)\setminus N(Z_i)\}$, in other words, C satisfies $\mathcal{C}(C, Z_i) = 0$

$L = \{u\in X_i\setminus A \mid u\in N(Z_j)\setminus N(Z_g)\}$, in other words, L satisfies $\mathcal{C}(L, Z_g) = 0$

$R = \{u\in X_i\setminus A \mid u\in N(Z_g\setminus N(Z_j)\}$, in other words, R satisfies $\mathcal{C}(R, Z_j) = 0$

$B = \{u\in X_i\setminus A \mid u\in N(Z_j)\cap N(Z_g)\}$.

Further, for the solution S, define the SN-function $f : X_i^+\to 2^V$ as follows:

$$\begin{aligned}
f(u) &= N(u)\setminus\{v\in N(u)\mid v\in\hat{Z}\}, \text{if } u\in X_i\\
&= N(u), \text{if } u\in X_i^+\setminus X_i
\end{aligned}$$

Note: f is dependent on S and i and wherever f is used, it must be used as per the definition at the corresponding node in the tree decomposition.

Let $S\subseteq V$ be a subset of size k, i be a node in H in the nice tree decomposition, f be the SN-function for X_i^+ defined using S, $\mathtt{P} = (A, C, L, R, B)$ be a partition of X_i, and S_i, Z_i and $\hat{Z}$ be as defined above. The following two lemmas are useful in setting up a recursive definition of $\mathcal{C}(V, S)$ and a bottom-up evaluation of the recurrence.

Lemma 3. $\mathcal{C}(V,S) = \mathcal{C}(V\setminus X_i^+, \hat{Z}\cup A) + \mathcal{C}(X_i\setminus A, \hat{Z}) + \mathcal{C}_f(X_i^+, S_i)$.

Proof. The expected coverage of V by the set S is given by $\mathcal{C}(V,S) = \mathcal{C}(V\setminus X_i^+, S) + \mathcal{C}(X_i, S) + \mathcal{C}(X_i^+\setminus X_i, S)$. Since V is partitioned into three disjoint sets X_i, $X_i^+\setminus X_i$, and $V\setminus X_i^+$, we get

$$\mathcal{C}(V,S) = \mathcal{C}(V\setminus X_i^+, \hat{Z}\cup A) + \mathcal{C}(X_i\setminus A, \hat{Z}\cup S_i) + \mathcal{C}(A,S) + \mathcal{C}(X_i^+\setminus X_i, S_i).$$

By applying Lemma 2 to $\mathcal{C}(X_i\setminus A, \hat{Z}\cup S_i)$, it follows that

$$\mathcal{C}(V,S) = \mathcal{C}(V\setminus X_i^+, \hat{Z}\cup A) + \mathcal{C}(X_i\setminus A, \hat{Z}) + \mathcal{C}_f(X_i\setminus A, S_i) + \mathcal{C}(A,S) + \mathcal{C}(X_i^+\setminus X_i, S_i).$$

Further, $\mathcal{C}(A, S_i) = \mathcal{C}(A, S)$, since $\mathcal{C}(A, S_i) = \mathcal{C}(A, S) = \sum_{v \in A} w(v)$, it follows that

$$\mathcal{C}(V, S) = \mathcal{C}(V \setminus X_i^+, \hat{Z} \cup A) + \mathcal{C}(X_i \setminus A, \hat{Z}) + \mathcal{C}_f(X_i \setminus A, S_i) + \mathcal{C}(A, S_i) + \mathcal{C}(X_i^+ \setminus X_i, S_i).$$

By the definition of f and $\mathcal{C}_f$, $\mathcal{C}_f(X_i \setminus A, S_i) + \mathcal{C}(A, S_i) + \mathcal{C}(X_i^+ \setminus X_i, S_i) = \mathcal{C}_f(X_i^+, S_i)$. Therefore we get $\mathcal{C}(V, S) = \mathcal{C}(V \setminus X_i^+, \hat{Z} \cup A) + \mathcal{C}(X_i, \hat{Z}) + \mathcal{C}_f(X_i^+, S_i)$. Hence the lemma. □

As a corollary it follows that $\mathcal{C}(V, S) = \mathcal{C}_f(X_r^+, S_r)$. Recall that $X_r = \emptyset$, $S_r = S$ and f is such that $f(u) = N(u)$ for all $u \in X_i^+$. We now show that for each node $i \in H$, $\mathcal{C}_f(X_i^+, S_i)$ can be written in terms of the appropriate sub-problems in children j and g of the node i.

Lemma 4.

$$\mathcal{C}_f(X_i^+, S_i) = w(A) + \mathcal{C}_f(C, A) + \mathcal{C}_f(L, A \cup Z_j) + \mathcal{C}_f(R, A \cup Z_g) + \mathcal{C}_f(B, S_i) + \mathcal{C}(X_i^+ \setminus X_i, S_i).$$

Proof. By definition, the expected coverage of X_i^+ by S_i restricted by f is given by the following equations.

$$\begin{aligned}
\mathcal{C}_f(X_i^+, S_i) &= \mathcal{C}_f(X_i, S_i) + \mathcal{C}_f(X_i^+ \setminus X_i, S_i) \\
&= w(A) + \mathcal{C}_f(C, S_i) + \mathcal{C}_f(L, S_i) + \mathcal{C}_f(R, S_i) + \mathcal{C}_f(B, S_i) + \mathcal{C}(X_i^+ \setminus X_i, S_i) \\
&= w(A) + \mathcal{C}_f(C, A) + \mathcal{C}_f(L, A \cup Z_j) + \mathcal{C}_f(R, A \cup Z_g) + \mathcal{C}_f(B, S_i) \\
&\quad + \mathcal{C}(X_i^+ \setminus X_i, S_i)
\end{aligned}$$

The second equation follows from the first due to the definition of the partition $\mathtt{P} = (A, C, L, R, B)$, and the fact that for all $u \in X_i^+ \setminus X_i$, $f(u) = N(u)$. The third equation follows from the second due to the fact that $\mathcal{C}_f(L, S_i) = \mathcal{C}_f(L, A \cup Z_j)$ and $\mathcal{C}_f(R, S_i) = \mathcal{C}_f(R, A \cup Z_g)$, since L and R are sets for which $\mathcal{C}(L, Z_g) = \mathcal{C}(R, Z_j) = 0$. □

Lemmas 3 and 4 show that the expected coverage of a set S of size k can be computed in a bottom-up manner over the nice tree decomposition. Further, at a node $i \in H$ the expected coverage of X_i^+ by a S is $\mathcal{C}(X_i, \hat{Z}) + \mathcal{C}_f(X_i^+, S_i)$, where f is uniquely determined by S, i and the tree decomposition. Also, $\mathcal{C}_f(X_i^+, S_i)$ is decomposed into six terms based on the partition $\mathtt{P}$ which is uniquely determined by S, i and the tree decomposition. Of these six terms, five of them are computed at the node $i \in H$, and one term comes from j and g, which are the children of i. Therefore, the search for the optimum S of size k can be performed in a bottom-up manner by enumerating all possible choices of the 5-way partition of X_i and all possible choices of the SN-function f at X_i. For each candidate partition of X_i and the SN-function f, the optimum S_i is computed by considering the *compatible* solutions at X_j and X_g. This completes the recursive formulation of the expected coverage of a set S of size k. We next present the bottom-up evaluation of the recurrence to compute the optimum set which is $\arg\max_{|S| \leq k} \mathcal{C}(V, S)$.

4.2. Bottom-Up Computation of an Optimal Set

For each node $i \in H$, we associate a table T_i. Each row in the table T_i is a triple which consists of an integer b, a five way partitioning $\mathtt{P} = (A, C, L, R, B)$ of X_i and an SN-function f defined on X_i^+. Throughout this section we assume that for vertices $u \in X_i^+ \setminus X_i$, $f(u) = N(u)$. The columns corresponding to a row $(b, \mathtt{P}, f)$ is a vertex set $T_i[b, \mathtt{P}, f].\mathtt{Solution}$ and a value $T_i[b, \mathtt{P}, f].\mathtt{Value}$. Let S denote $T_i[b, \mathtt{P}, f].\mathtt{Solution}$. To define the value $T_i[b, \mathtt{P}, f].\mathtt{Value}$ associated with S, consider $Z = S \setminus A$, and let $Z_j = Z \cap (X_j^+ \setminus X_j)$ and $Z_g = Z \cap (X_g^+ \setminus X_g)$. Then, $T_i[b, \mathtt{P}, f].\mathtt{Value} = w(A) + \mathcal{C}_f(C, A) + \mathcal{C}_f(L, A \cup Z_j) + \mathcal{C}_f(R, A \cup Z_g) + \mathcal{C}_f(B, A \cup Z) + \mathcal{C}(X_i^+ \setminus X_i, S)$. The set $T_i[b, \mathtt{P}, f].\mathtt{Solution}$ is a set S of size b such that $S \cap X_i = A$ and the associated value $T_i[b, \mathtt{P}, f].\mathtt{Value}$ is maximized.

Leaf node. Let $i \in H$ be a leaf node with bag $X_i = \emptyset$. The only possible five-way partition of an empty set is the set with five empty sets and the budget is $b = 0$. The only valid SN-function is $f : \emptyset \to \emptyset$. Therefore, the value of the corresponding row in the table is

$$T_i[b, \mathtt{P}, f] = \{\mathtt{Solution} = \emptyset, \mathtt{Value} = 0\} \text{ for } b = 0, \mathtt{P} = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \text{ and } f : \emptyset \to \emptyset.$$

It is clear that the empty set is the set that achieves the maximum for the MAX-EXP-COVER-1-RF problem on the empty graph with budget $b = 0$. Therefore, at the leaf nodes in H, the table T_i maintains the optimum solution for each row. The time to update an entry is $\mathcal{O}(1)$.

Introduce node. Let i be an introduce node with child j such that $X_i = X_j \cup \{v\}$ for some $v \notin X_j$. Let $0 \le b \le k$ be an integer, $\mathtt{P} = (A, C, L, R, B)$ be a five-way partition of X_i and f be the SN-function defined on X_i^+. The computation of the table entry is split into two cases, depending on whether the vertex v belongs to the set A in the partition $\mathtt{P}$ or not.

1. **Case** $v \in A$**.** Define $C_v = \{u \in X_i \setminus A \mid v \in N(u) \cap f(u)\}$. Let $\mathtt{P}_j$ denote the partition of X_j obtained by removing vertex v from the set A of the partition $\mathtt{P}$. Let $f_j : X_j \to 2^V$ be the SN-function defined as follows:

$$f_j(u) = \begin{cases} f(u) \setminus \{v\} & \text{if } u \in C_v \\ f(u) & \text{otherwise.} \end{cases}$$

   Then,

$$\begin{aligned} T_i[b, \mathtt{P}, f].\mathtt{Solution} &= T_j[b-1, \mathtt{P}_j, f_j].\mathtt{Solution} \cup \{v\} \\ T_i[b, \mathtt{P}, f].\mathtt{Value} &= T_j[b-1, \mathtt{P}_j, f_j].\mathtt{Value} + w(v) + \mathcal{C}_f(C_v, v) \end{aligned}$$

2. **Case** $v \notin A$**.** Since v is in X_i but not in X_j it follows that $N(v) \cap X_i^+ \subseteq X_i$. Therefore, the coverage of the vertex v by the vertices that occur only in $X_j^+ \setminus X_j$ is zero. Let $\mathtt{P}_j$ be the partition of X_j obtained by removing the vertex v from the appropriate set in the partition $\mathtt{P}$. The SN-function f_j is defined as follows on the set X_j^+: For $u \in X_i \setminus \{v\}$, $f_j(u) = f(u)$.

$$\begin{aligned} T_i[b, \mathtt{P}, f].\mathtt{Solution} &= T_j[b, \mathtt{P}_j, f_j].\mathtt{Solution} \\ T_i[b, \mathtt{P}, f].\mathtt{Value} &= T_j[b, \mathtt{P}_j, f_j].\mathtt{Value} + \mathcal{C}_f(v, A) \end{aligned}$$

Forget node. Let i be a forget node with child j such that $X_i = X_j \setminus \{v\}$ for some $v \in X_j$. Let $0 \le b \le k$ be a budget, $\mathtt{P} = (A, C, L, R, B)$ be a five-way partition of X_i and f be an SN-function. We consider the following five-way partitions of X_j.

- $\mathtt{P}_1 = (A \cup \{v\}, C, L, R, B)$
- $\mathtt{P}_2 = (A, C \cup \{v\}, L, R, B)$
- $\mathtt{P}_3 = (A, C, L \cup \{v\}, R, B)$
- $\mathtt{P}_4 = (A, C, L, R \cup \{v\}, B)$
- $\mathtt{P}_5 = (A, C, L, R, B \cup \{v\})$

Let f_j be the SN-function defined as follows: for $u \in X_i$, $f_j(u) = f(u)$ and $f_j(v) = N(v)$. Let $\mathtt{P}_j = \underset{\mathtt{P}' \in \{\mathtt{P}_i\}_{i=1}^5}{\arg\max}\, T_j[b, \mathtt{P}', f_j].\mathtt{Value}$. $T_i[b, \mathtt{P}, f]$ is defined as follows:

$$\begin{aligned} T_i[b, \mathtt{P}, f].\mathtt{Solution} &= T_j[b, \mathtt{P}_j, f_j].\mathtt{Solution} \\ T_i[b, \mathtt{P}, f].\mathtt{Value} &= T_j[b, \mathtt{P}_j, f_j].\mathtt{Value} \end{aligned}$$

Join node. Let i be a join node with children j and g such that $X_i = X_j = X_g$. Let $0 \leq b \leq k$ be a budget, $\mathtt{P} = (A, C, L, R, B)$ be a five-way partition of X_i and f be an SN-function. We consider the sets $\mathcal{U}_j$ and $\mathcal{U}_g$ of all SN-functions defined on X_j^+ and X_g^+, respectively, satisfying the following conditions:

- For each $v \in L$, $f_j(v) = f(v)$ and $f_g(v) = \emptyset$.
- For each $v \in R$, $f_j(v) = \emptyset$ and $f_g(v) = f(v)$.
- For each $v \in B$, $f_j(v)$ and $f_g(v)$ are defined as follows. For each partition $f_1(v) \cup f_2(v) = f(v) \cap X_i^+$, $f_j(v) = f_1(v) \cup (f(v) \setminus X_i^+)$ and $f_g(v) = f_2(v) \cup (f(v) \setminus X_i^+)$.

We then consider all possible candidates for L_j, R_j, B_j such that $L_j \cup R_j \cup B_j = L \cup B$ and L_g, R_g, B_g such that $L_g \cup R_g \cup B_g = R \cup B$. Each such candidate defines a $\mathtt{P}'_j = (A, C \cup R, L_j, R_j, B_j)$ of X_j and a $\mathtt{P}'_g = (A, C \cup L, L_g, R_g, B_g)$ of X_g. Let $\mathcal{R}_j$ and $\mathcal{R}_g$ denote the set of all such candidate partitions of X_j and X_g, respectively.

Let $\mathtt{P}_j \in \mathcal{R}_j$, $\mathtt{P}_g \in \mathcal{R}_g$, $f_j \in \mathcal{U}_j$, $f_g \in \mathcal{U}_g$ and $0 \leq b' \leq b - |A|$ be the values at which the maximum value for Equation (2) is achieved.

$$\max_{\substack{0 \leq b_j \leq b - |A| \\ \mathtt{P}'_j \in \mathcal{R}_j, \mathtt{P}'_g \in \mathcal{R}_g \\ f'_j \in \mathcal{U}_j, f'_g \in \mathcal{U}_g}} T_j[b_j + |A|, \mathtt{P}'_j, f'_j].\mathtt{Value} + T_g[b - b_j, \mathtt{P}'_g, f'_g].\mathtt{Value}. \tag{2}$$

The value of the row corresponding to $(b, \mathtt{P}, f)$ in T_i is given as follows:

$$\begin{aligned} T_i[b, \mathtt{P}, f].\mathtt{Solution} &= T_j[b' + |A|, \mathtt{P}_j, f_j].\mathtt{Solution} \cup T_g[b - b', \mathtt{P}_g, f_g].\mathtt{Solution} \\ T_i[b, \mathtt{P}, f].\mathtt{Value} &= T_j[b' + |A|, \mathtt{P}_j, f_j].\mathtt{Value} + T_g[b - b', \mathtt{P}_g, f_g].\mathtt{Value} - \mathcal{C}_f(A \cup C, A). \end{aligned}$$

We now prove that the update steps presented above are correct.

Lemma 5. *For each node $i \in H$, for each row $(b, \mathtt{P}, f)$ in T_i, the pair $T_i[b, \mathtt{P}, f].\mathtt{Solution}$, $T_i[b, \mathtt{P}, f].\mathtt{Value}$ is such that $\mathcal{C}(X_i^+, T_i[b, \mathtt{P}, f].\mathtt{Solution}) = T_i[b, \mathtt{P}, f].\mathtt{Value}$, and this is the maximum possible value.*

Proof. The proof is by induction on the height of a node in H. The height of a node i in a rooted tree H is the distance to the farthest leaf in the subtree rooted at i. The base case is when i is a leaf node in H, and clearly its height is 0. For a leaf node i with $X_i = \emptyset$, the row with $b = 0$, $\mathtt{P} = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ and $f : \emptyset \to \emptyset$ is the only valid row entry and its value is 0. This completes the proof of the base case. Let us assume that the claim is true for all nodes of height at most $h - 1 \geq 0$. We prove that if the claim is true at all nodes of height at most $h - 1$, then it is true for a node of height h. Let b be a budget, $\mathtt{P} = (A, C, L, R, B)$ be a partition of X_i and f be an SN-function on X_i^+. Let $T_i[b, \mathtt{P}, f].\mathtt{Solution} = S = A \cup Z$. For any optimal $S' = A \cup Z'$ where $|S'| = b$, we prove that

$$T_i[b, \mathtt{P}, f].\mathtt{Value} \geq w(A) + \mathcal{C}_f(C, A) + \mathcal{C}_f(L, A \cup Z'_j) + \mathcal{C}_f(R, A \cup Z'_g) + \mathcal{C}_f(B, S') + \mathcal{C}(X_i^+ \setminus X_i, S').$$

We proceed by considering the type of node i so that the induction hypothesis can be applied at nodes of height at most $h - 1$.

Case when i is an introduce node. Let j be the child of i and $X_i = X_j \cup \{v\}$ for some $v \notin X_j$. We now consider two cases depending on whether v belongs to A or not.

1. **Case** $v \in A$. Let $\mathtt{P}_j$ be the partition of X_j obtained from $\mathtt{P}$ by removing v from A. Let f_j be the SN-function on X_j^+ such that $f_j(u) = f(u) \setminus v$, if $u \in C_v$ and $f_j(u) = f(u)$, otherwise. We know from our claimed optimality of S', and that $S' \cap X_i = S \cap X_i = A$, and the value of $T_i[b, \mathtt{P}, f]$, that

$$\begin{aligned} \mathcal{C}_f(X_i^+, S') &= \mathcal{C}_{f_j}(X_j^+, S' \setminus \{v\}) + \mathcal{C}_f(C_v, v) \\ &> \mathcal{C}_f(X_i^+, S) = \mathcal{C}_{f_j}(X_j^+, S \setminus \{v\}) + \mathcal{C}_f(C_v, v). \end{aligned}$$

Therefore, it follows that $\mathcal{C}_{f_j}(X_j^+, S' \setminus \{v\}) > \mathcal{C}_{f_j}(X_j^+, S \setminus \{v\}) = T_j[b-1, \mathtt{P}_j, f_j]\texttt{.Value}$. In other words, we have concluded that the value for the row $(b-1, \mathtt{P}_j, f_j)$ in T_j is not the optimum value. This contradicts our premise at node j, which is of height at most $h-1$ for which, by induction hypothesis, the table maintains the optimal values. Therefore, our assumption that $T_i[b, \mathtt{P}, f]$ is not optimum is wrong.

2. **Case $v \notin A$.** Let $\mathtt{P}_j$ be the partition of X_j obtained by removing v from the appropriate set in the partition $\mathtt{P}$. Let f_j be the SN-function on X_j^+ such that $f_j(u) = f(u)$ for each $u \in X_j^+$. We know from our claimed optimality of S', and that $S' \cap X_i = S \cap X_i = A$, and the value of $T_i[b, \mathtt{P}, f]$ that $\mathcal{C}_f(X_i^+, S') = \mathcal{C}_{f_j}(X_j^+, S') + \mathcal{C}_f(v, A) > \mathcal{C}_f(X_i^+, S) = \mathcal{C}_{f_j}(X_j^+, S) + \mathcal{C}_f(v, A)$. Therefore, it follows that $\mathcal{C}_{f_j}(X_j^+, S') > \mathcal{C}_{f_j}(X_j^+, S)$. In other words, we have concluded that the value for the row $(b, \mathtt{P}_j, f_j)$ in T_j is not the optimum value. This contradicts our premise at node j, which is of height at most $h-1$ for which, by induction hypothesis, the table maintains the optimal values. Therefore, our assumption that $T_i[b, \mathtt{P}, f]$ is not optimum is wrong.

Forget node. We know that $X_i^+ = X_j^+$, and v is in X_j but not in X_i, it follows that $N(v) \cap X_i = \emptyset$ and $N(v) \subseteq X_j^+$. Define f_j to be the SN-function at X_j^+ such that $f_j(u) = f(u)$ for each $u \in X_i$ and $f_j(v) = N(v)$. We have assumed $T_i[b, \mathtt{P}, f]\texttt{.Value} < \mathcal{C}_f(X_i^+, S')$. Further, since $X_i^+ = X_j^+$ and due to the definition of f_j, $\mathcal{C}_f(X_i^+, S') = \mathcal{C}_{f_j}(X_j^+, S')$. Since $T_i[b, \mathtt{P}, f]$ is computed identically from some row in T_j, let us say $(b, \mathtt{P}_j, f_j)$, it follows that $T_j[b, \mathtt{P}_j, f_j]\texttt{.Value} < \mathcal{C}_{f_j}(X_j^+, S')$. This contradicts the premise that the table T_i is at the lowest height in the tree decomposition at which an entry is sub-optimal. Therefore, our premise is wrong.

Join node. We assume that S' is indeed a better solution than S for the table entry $(b, \mathtt{P}, f)$ of T_i. Let $S'_j = S \cap (X_j^+ \setminus X_j)$ and $S'_g = S \cap (X_g^+ \setminus X_g)$. Let $b'_j = |S'_j|$. Let $\mathtt{P}'_j = (A, C \cup R, L'_j, R'_j, B'_j)$ and $\mathtt{P}'_g = (A, C \cup L, L'_g, R'_g, B'_g)$ be the partitions of X_j and X_g defined using S'_j and S'_g, respectively. Note that, $L'_j \cup R'_j \cup B'_j = L \cup B$ and $L'_g \cup R'_g \cup B'_g = R \cup B$. Let f'_j be the SN function on X_j^+ such that $f'_j(u) = f(u)$ for $u \in A \cup C \cup L \cup B$ and $f'_j(u) = \emptyset$ for $u \in R$. Let f'_g be the SN function on X_g^+ such that $f'_g(u) = f(u)$ for $u \in A \cup C \cup R$, $f'_g(u) = \emptyset$ for $u \in L$ and $f'_g(u) = f(u) \setminus S'_j$. The coverage $\mathcal{C}_f(X_i^+, S')$ can be written as $\mathcal{C}_f(X_i^+, S') = \mathcal{C}_{f'_j}(X_j^+, S'_j) + \mathcal{C}_{f'_g}(X_g^+, S'_g) - w(A) - \mathcal{C}_f(C, A)$, where the coverage of X_j^+ by S'_j and X_g^+ by S'_g are restricted to the partitions $\mathtt{P}'_j$ and $\mathtt{P}'_g$.

We know that $X_i = X_j = X_g$ in case of join node. The table entry $(b, \mathtt{P}, f)$ is updated using the entries $(b_j + |A|, \mathtt{P}_j, f_j)$ and $(b - b_j, \mathtt{P}_g, f_g)$ of the table T_j and T_g, respectively. In other words, the values of variables $b_j, \mathtt{P}_j, \mathtt{P}_g, f_j$ and f_g are obtained from the Equation (2) described in the recursive definition of join node. The values $b'_j, \mathtt{P}'_j, \mathtt{P}'_g, f'_j$ and f'_g are also feasible for the range given in the Equation (2). Then, we have $T_j[b'_j + |A|, \mathtt{P}'_j, f'_j]\texttt{.Value} + T_g[b - b'_j, \mathtt{P}'_g, f'_g]\texttt{.Value} < T_j[b' + |A|, \mathtt{P}_j, f_j]\texttt{.Value} + T_g[b - b', \mathtt{P}_g, f_g]\texttt{.Value}$. Since S' is better than S for the entry $(b, \mathtt{P}, f)$ of T_i, we have $T_j[b' + |A|, \mathtt{P}_j, f_j]\texttt{.Value} + T_g[b - b', \mathtt{P}_g, f_g]\texttt{.Value} < \mathcal{C}_{f'_j}(X_j^+, S'_j) + \mathcal{C}_{f'_g}(X_g^+, S'_g)$. The above inequality shows that at least one of the table entries $T_j[b'_j + |A|, \mathtt{P}'_j, f'_j]$ or $T_g[b - b'_j, \mathtt{P}'_g, f'_g]$ is not optimal. This would again contradict the premise that i is the node at the least height at which some table entry is sub-optimal.
This completes the case analysis and our proof. Hence the lemma. □

Running Time. There are $\mathcal{O}(n\, tw)$-many nodes in the nice tree decomposition H. Each node $i \in H$ has a maximum of $(k+1)(5 \cdot 2^{\Delta})^{tw}$ entries. The $2^{\Delta \cdot tw}$ comes from the fact that at each vertex in a bag, we enumerate all subsets of neighbors to come up with the SN-functions. It is clear from the description that at the leaf nodes, introduce nodes, and forget nodes, the update time is $\mathcal{O}(tw)$. At a join node, the time taken to compute an entry depends on three basic operations. The optimal partitions $\mathtt{P}_j$ and $\mathtt{P}_g$ are computed in $3^{|L|+|R|+2|B|}$ time and budget distribution can be done in $\mathcal{O}(k)$ time. The costliest operation is to enumerate the different SN-functions f_j and f_g for the given SN-function f. Since we need to consider all the 2^{Δ}-possible ways of distributing the $f(v)$ for each vertex v, the distribution takes $\mathcal{O}((2^{\Delta})^{|B|})$ time. Therefore, the running time for an entry $(b, \mathtt{P}, f)$ in a join node takes $\mathcal{O}(k \cdot 3^{|L|+|R|+2|B|} \cdot (2^{\Delta})^{|B|})$ time, and this is $2^{\mathcal{O}(\Delta \cdot tw)}$. This analysis of the running time and Lemma 5 complete the proof of the following theorem which is our main result.

Theorem 1. *The* MAX-EXP-COVER-1-RF *problem can be solved in time* $2^{\mathcal{O}(\Delta \cdot tw)} n^{\mathcal{O}(1)}$.

5. Parameterized Complexity of PROBABILISTIC-CORE Problem

Let $\mathcal{G} = (V, E, p)$ be an uncertain graph and d be an integer. Given a set $K \subseteq V$, we define the probability that the set K is being a d-core in $\mathcal{G}$, denoted by $\rho_d(\mathcal{G}, K)$, to be $\sum_{H \sqsubseteq \mathcal{G}} P(H) I(H, K, d)$, where I is an indicator variable that takes value 1 if and only if the set K forms a d-core in the graph H. The decision version of the PROBABILISTIC-CORE problem is formally stated as follows: given an uncertain graph $\mathcal{G} = (V, E, p)$, an integer d and a probability $\theta \in [0, 1]$, decide if there exists a set $K \subseteq V$ such that $\rho_d(\mathcal{G}, K) \geq \theta$.

We study the PROBABILISTIC-CORE problem under the LRO and RF model. First we show that the PROBABILISTIC-CORE-LRO problem and the INDIVIDUAL-CORE problem are polynomial time solvable, due to the fact that an uncertain graph under the LRO model has only a polynomial number of worlds. Then we show that the PROBABILISTIC-CORE-RF problem is W[1]-hard for the parameter d and admits an FPT algorithm for the parameter treewidth.

5.1. *An Exact Algorithm for the* PROBABILISTIC-CORE-LRO *Problem*

We present a polynomial time algorithm (see as Algorithm 1) for the PROBABILISTIC-CORE-LRO problem. Let $\langle \mathcal{G} = (V, E, p), d \rangle$ be an instance of the PROBABILISTIC-CORE-LRO problem. As per the definition of the LRO model, let the set $\{G_0, \ldots, G_m\}$ be the possible worlds of $\mathcal{G}$, where $G_0 = (V, \emptyset)$ and for $0 \leq i \leq m-1$, $E(G_{i+1}) = E(G_i) \cup \{e_i\}$. For each $0 \leq i \leq m-1$, we consider the linear order in which G_i precedes G_{i+1}.

Algorithm 1: PROBABILISTIC-CORE-LRO

```
Data: $\mathcal{G} = (V, E, p), d$
Result: A set $C \subseteq V$ with a probability $q$.
for $i \leftarrow 0$ to $m$ do
    $C \leftarrow V$;
    while $\delta(G_i[C]) < d$ do
        Let $v$ be a vertex such that $deg_{G_i[C]}(v) < d$;
        $C \leftarrow C \setminus \{v\}$ ;
    end
    if $C \neq \emptyset$ then
        return $C, p(e_i)$;
    end
end
return $\emptyset, 0.0$;
```

Lemma 6. *The Algorithm 1 solves the* PROBABILISTIC-CORE-LRO *problem in polynomial time.*

Proof. By definition of the possible worlds, $G_m = (V, E)$. If $\delta(G_m) < d$, then G_m does not contain a d-core. Moreover, for $0 \leq i < m$, since G_i is an edge subgraph of G_m, no graph in the possible worlds has a d-core. This is indicated by the return value-pair of the $\emptyset$ and probability 0.0. On the other hand if $\delta(G_m) \geq d$, let G_i be the first graph in the linear ordering of possible worlds for which a non-empty C is computed by Algorithm 1 on exit from the While-loop. Clearly, C induces a d-core in G_i. Since G_i is the first graph which contains a d-core, for each $0 \leq j \leq i-1$, G_j does not contain a d-core. Further, for every $j > i$, C is a d-core in G_j since it is an edge-subgraph of G_i. Then,

$$\rho_d(\mathcal{G},C) = \sum_{j=0}^{m} P(G_j)I(G_j,C,d) = \sum_{j=i}^{m} P(G_j) = p(e_i).$$

For any set $K \subseteq V$ with $\rho_d(\mathcal{G},K) > 0$, the set K can form a d-core only in the possible worlds $\{G_i, G_{i+1}, \ldots, G_m\}$. Thus, $\rho_d(\mathcal{G},K) \leq \rho_d(\mathcal{G},C)$. This completes the proof. □

The following observation again uses the fact that there are only a polynomial number of possible worlds under the LRO model.

Observation 3. *The* INDIVIDUAL-CORE *problem is polynomial time solvable on uncertain graphs under the LRO model.*

Proof. Clearly, for each $0 \leq i \leq m$, the While-loop in Algorithm 1 computes the maximal d-core in G_i. Therefore, for each v, $q_d(v) = p(e_i)$, where i the smallest index such that the non-empty d-core contains v. Since the number of possible worlds are $m+1$, for each vertex $v \in V$, the probability $q_d(v)$ can be computed in polynomial time. Thus, for a given uncertain graph $\mathcal{G} = (V,E,p)$, an integer d and probability threshold θ, the set $\{v \mid q_d(v) \geq \theta\}$ is the optimum solution to the INDIVIDUAL-CORE problem. □

5.2. *Parameterized Complexity of the* PROBABILISTIC-CORE-RF *Problem*

We show that the PROBABILISTIC-CORE-RF problem is W[1]-hard. The reduction that we show for the W[1]-hardness is similar to the hardness result of the INDIVIDUAL-CORE problem shown by Peng et al. [16] (they call this the (d,θ)-CORE problem). The reduction is from the CLIQUE problem which is as follows: given a graph $G = (V,E)$ and an integer k, decide if G has a clique of size at least k. The CLIQUE problem is known to be W[1]-hard [47].

Theorem 2. *The* PROBABILISTIC-CORE-RF *problem is W[1]-hard for the parameter d.*

Proof. Let $\langle G = (V,E), k\rangle$ be an instance of the CLIQUE problem. The output of the reduction is denoted by $\langle \mathcal{G} = (V,E,p), d, \theta\rangle$, and it is an instance of PROBABILISTIC-CORE-RF problem. The vertex set and edge set of $\mathcal{G}$ are same as V and E, respectively, $d = k-1$ and $\theta = 2^{-\binom{k}{2}}$. Further, for each edge $e \in E$, define $p(e) = \frac{1}{2}$ in $\mathcal{G}$. Now we show that the CLIQUE problem on the instance $\langle G,k\rangle$ and the PROBABILISTIC-CORE-RF problem on the instance $\langle \mathcal{G},d,\theta\rangle$ are equivalent.

We prove the forward direction first. Let $K \subseteq V$ be a k-clique in G. The set K is indeed a d-core since every vertex $v \in K$ has $k-1$ neighbors in K. The probability that K is a d-CORE in a random sample from the possible worlds of $\mathcal{G}$ is $2^{-\binom{k}{2}}$. Thus, the set K is a feasible solution for the instance $\langle \mathcal{G},d,\theta\rangle$ of the PROBABILISTIC-CORE-RF problem.

Now we prove the reverse direction. We claim that any feasible solution $K \subseteq V$ contains exactly $d+1$ vertices. If K has less than $d+1$ vertices then K cannot form a d-core in any possible world. Therefore, we consider the case in which that K has more than $d+1$ vertices, and each vertex in the set K has degree at least d. Consequently, the number of edges in any possible world in which K is a d-core is at least $\frac{d(d+2)}{2} = \frac{d(d+1)}{2} + \frac{d}{2} = \binom{d+1}{2} + \frac{d}{2}$. Then, the probability that the set K is a d-core in $\mathcal{G}$ is at most $2^{-\binom{d+1}{2}-\frac{d}{2}} < \theta$, and this contradicts the hypothesis that K is a feasible solution. It follows that any feasible solution $K \subseteq V$ contains exactly $d+1$ vertices and each vertex has degree d, thus K is a k clique. Hence the theorem. □

5.3. *The* PROBABILISTIC-CORE-RF *Problem is FPT by Treewidth*

We show that the PROBABILISTIC-CORE-RF problem admits an FPT algorithm with treewidth as the parameter. The input consists of an instance $\langle \mathcal{G} = (V,E,p), d\rangle$ of the PROBABILISTIC-CORE-RF problem. A nice tree decomposition $(X,\mathcal{T})$ of the graph $G = (V,E)$ is also given as part of the input.

Let i be any node in $\mathcal{T}$ and $\alpha : X_i \rightarrow \{0, 1, \ldots, d\}$ and $\beta : X_i \rightarrow \{0, 1, \ldots, d\}$ be a pair of functions on X_i. Given a set $K \subseteq V$, the set K is said to be (α, β)-constrained d-core of G if

1. for each $v \in X_i \cap K$, $|N(v) \cap K| = \alpha(v) + \beta(v)$ and $|N(v) \cap K \cap X_i| = \alpha(v)$, and
2. for each $v \in K \setminus X_i$, $|N(v) \cap K| = d$.

Let $\mathcal{G}_i^{\alpha,\beta}$ denote the uncertain graph $\mathcal{G}[X_i^+ \setminus (\alpha^{-1}(0) \cap \beta^{-1}(0))]$. We define a constrained version of the PROBABILISTIC-CORE-RF problem as follows. Given a set K, the probability that the set K is an (α, β)-constrained d-core in $\mathcal{G}_i^{\alpha,\beta}$ is given by:

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, K) = \sum_{H \sqsubseteq \mathcal{G}_i^{\alpha,\beta}} P(H) I'(H, \alpha, \beta, K, d),$$

where $I'(H, \alpha, \beta, K, d)$ is an indicator function that takes value 1 if and only if K is an (α, β)-constrained d-core of H. The optimization version of the (α, β)-constrained PROBABILISTIC-CORE-RF problem seeks to find the set K such that $\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, K)$ is maximized. The solution for (α, β)-constrained PROBABILISTIC-CORE-RF problem for different values of i, α and β on $\mathcal{G}$ are partial solutions which are used to come up with a recursive specification of the optimum value. The dynamic programming (DP) formulation on the nice tree decomposition $(X, \mathcal{T})$ results in an FPT algorithm with treewidth as the parameter.

The dynamic programming formulation maintains a table T_i at every node $i \in X$. Each row in the table T_i is a pair of functions α and β such that $\alpha, \beta : X_i \rightarrow \{0, 1, \ldots, d\}$. The column corresponding to the row (α, β) is a pair `(Solution,Value)` which consists of a set $S_{\alpha,\beta}$ and a probability value. Further, the set $S_{\alpha,\beta}$ is an optimal solution for the (α, β)-constrained PROBABILISTIC-CORE-RF problem on the instance $\langle \mathcal{G}_i^{\alpha,\beta}, d \rangle$.

Intuitively, the functions α and β defined on X_i stand for the "in-bag-degree" and "out-bag-degree" constraints, respectively, for each vertex $v \in X_i$. We maintain the candidate solutions in the table for different values of the functions α and β. We allow the candidate solutions that are infeasible at current stage and those vertices which are not satisfied with degree d will get neighbors from nodes at a higher level in the tree decomposition. The optimal solution for the instance $\langle \mathcal{G}, d \rangle$ of the PROBABILISTIC-CORE-RF problem can be obtained from the table T_r where r is the root of the tree decomposition $\mathcal{T}$.

5.3.1. Dynamic Programming

We now present the dynamic programming formulation on the different types of nodes in $\mathcal{T}$. Let $i \in \mathcal{T}$ be a node with bag X_i. For a pair of functions $\alpha, \beta : X_i \rightarrow \{0, 1, \ldots, d\}$, we show how to compute the table entry $T_i[\alpha, \beta]$ in each type of node as follows.

Leaf node. Let i be a leaf node with bag $X_i = \emptyset$. We have one row in the table T_i. Let $\alpha, \beta : \emptyset \rightarrow \{0\}$ be the pair of functions corresponding to the row and the value of the table entry is given as:

$$T_i[\alpha, \beta] = (\emptyset, 0.0).$$

Insert node. Let i be an insert node with a child j, and let $X_i = X_j \cup \{v\}$ for some $v \notin X_j$. The row $T_i[\alpha, \beta]$ is computed based on the value of $\alpha(v)$ and $\beta(v)$. If $\beta(v) > 0$, then the row $T_i[\alpha, \beta]$ becomes infeasible since $N(v) \cap X_i^+ \subseteq X_i$. That is,

$$T_i[\alpha, \beta] = (\emptyset, 0.0).$$

In the rest of the cases we have $\beta(v) = 0$. We define $\beta' : X_j \rightarrow \{0, 1, \ldots, d\}$ to be a function such that $\beta'(u) = \beta(u)$ for all $u \in X_j$. When $\alpha(v) = 0$, then the vertex v will not be in the solution $S_{\alpha,\beta}$. Let $\alpha' : X_j \rightarrow \{0, 1, \ldots, d\}$ be a function such that $\alpha'(u) = \alpha(u)$ for all $u \in X_j$. The value of the row $T_i[\alpha, \beta]$ is same as $T_j[\alpha', \beta']$ since v is excluded. When $\alpha(v) > 0$, then v is part of $S_{\alpha,\beta}$.

Let $U = (N(v) \cap X_i) \setminus \alpha^{-1}(0)$. Then, we have $|U| \geq \alpha(v)$. Otherwise, no feasible solution exists for the row $T_i[\alpha, \beta]$. That is, degree constraint of v will not be met in any feasible solution. Assume $|U| \geq \alpha(v)$. Let $Y \subseteq U$ be a subset of U of size $\alpha(v)$, and $q(v, Y) = \prod_{u \in Y} p(uv)$. The vertices in Y that are the neighbors of v contribute degree $\alpha(v)$ to v. Then, each vertex $u \in Y$ will lose a degree from $\alpha(u)$ in the node j. Define $\alpha_Y : X_j \to \{0, 1, \ldots, d\}$ to be

$$\alpha_Y(u) = \begin{cases} \alpha(u) & \text{if } u \notin Y \\ \alpha(u) - 1 & \text{if } u \in Y. \end{cases}$$

Let

$$W = \underset{Y \subseteq U, |Y| = \alpha(v)}{\arg\max} \; q(v, Y) T_j[\alpha_Y, \beta'].\texttt{Value}$$

be the best $\alpha(v)$ neighbors of v such that the solution obtained from the table T_j is maximized. Then, the recursive definition of the row $T_i[\alpha, \beta]$ is given as,

$$T_i[\alpha, \beta] = (T_j[\alpha_W, \beta'].\texttt{Solution} \cup \{v\}, q(v, W) T_j[\alpha_W, \beta'].\texttt{Value}).$$

Forget node. Let i be a forget node with a child j, and let $X_i = X_j \setminus \{v\}$ for some $v \in X_j$. From the definition of the tree decomposition, it follows that $N(v) \subseteq X_j^+$. Since $N(v) \subseteq X_j^+$, either v is part of solution with constraint $\alpha(v) + \beta(v) = d$ or v is not part of solution. Let $U = (N(v) \cap X_j) \setminus \alpha^{-1}(0)$. For each $0 \leq a \leq \min(d, |U|)$ and $Y \subseteq U$ of size a, we define $\alpha_{a,Y}, \beta_{a,Y} : X_j \to \{0, 1, \ldots, d\}$ such that

$$\alpha_{a,Y}(u) = \begin{cases} \alpha(u) & \text{if } u \neq v \text{ and } u \notin Y \\ \alpha(u) + 1 & \text{if } u \neq v \text{ and } u \in Y \\ a & \text{if } u = v, \end{cases}$$

and

$$\beta_{a,Y}(u) = \begin{cases} \beta(u) & \text{if } u \neq v \text{ and } u \notin Y \\ \beta(u) - 1 & \text{if } u \neq v \text{ and } u \in Y \\ d - a & \text{if } u = v, \end{cases}$$

For $0 \leq a \leq min(d, |U|)$, let

$$\alpha_a, \beta_a = \underset{\alpha_{a,Y}, \beta_{a,Y} : Y \subseteq U, |Y| = a}{\arg\max} \; T_j[\alpha_{a,Y}, \beta_{a,Y}].\texttt{Value}.$$

Let $t = \underset{0 \leq a \leq min(d,|U|)}{\arg\max} \; T_j[\alpha_a, \beta_a].\texttt{Value}$. Then the value of the row $T_i[\alpha, \beta]$ is equal to $T_j[\alpha_t, \beta_t]$.

Join node. Let i be a join node with children j and g such that $X_i = X_j = X_g$. For a function α defined on X_i, we consider the function α for the child node X_j and for the other child node X_g, consider the function $\alpha' : X_i \to \{0, 1, \ldots, d\}$ such that $\alpha(u) = 0$ for all $u \in X_i$. Since β considers the neighbors from outside X_i, each vertex $v \in X_i$ with $\beta(v) > 0$ will get the d-core neighbors from the set $X_i^+ \setminus X_i$. Since $X_i^+ \setminus X_i = (X_j^+ \setminus X_j) \cup (X_g^+ \setminus X_g)$ and both the sets are disjoint, we divide $\beta(v)$ into two parts. For each vertex $v \in X_i$, we try all possible ways of dividing $\beta(v)$ into two parts. For $x : X_i \to \{0, 1, \ldots, d\}$ such that for each $v \in X_i$, $0 \leq x(v) \leq \beta(v)$, we define $\beta_x : X_j \to \{0, 1, \ldots, d\}$ and $\beta'_x : X_g \to \{0, 1, \ldots, d\}$ to be $\beta_x(v) = x(v)$ and $\beta'_x(v) = \beta(v) - x(v)$. Let

$$y = \underset{x : X_i \to \{0,1,\ldots,d\} | \forall v \in X_i, 0 \leq x(v) \leq \beta(v)}{\arg\max} \; T_j[\alpha, \beta_x].\texttt{Value} \cdot T_g[\alpha', \beta'_x].\texttt{Value}$$

Then, the recursive definition of the row $T_i[\alpha, \beta]$ is given as,

$$T_i[\alpha, \beta] = (T_j[\alpha, \beta_y].\texttt{Solution} \cup T_g[\alpha', \beta'_y].\texttt{Solution}, T_j[\alpha, \beta_y].\texttt{Value} \cdot T_g[\alpha', \beta'_y].\texttt{Value}).$$

5.3.2. Correctness and Running Time

Lemma 7. *Let i be a node in $\mathcal{T}$. For every pair of functions $\alpha, \beta : X_i \to \{0, 1, \ldots, d\}$, the row $T_i[\alpha, \beta]$ is computed optimally.*

Proof. Let i be a node in $\mathcal{T}$. We claim that for each pair of functions $\alpha, \beta : X_i \to \{0, 1, \ldots, d\}$, the row $T_i[\alpha, \beta]$ computes the optimal solution for the instance $\mathcal{G}_i^{\alpha,\beta}$ of the (α, β)-constrained PROBABILISTIC-CORE problem. That is, the set $S_{\alpha,\beta} = T_i[\alpha, \beta].\texttt{Solution}$ is an optimal solution for the above instance. Let $A = X_i \setminus (\alpha^{-1}(0) \cap \beta^{-1}(0))$. We show that for any $S = A \cup Z$ for some $Z \subseteq X_i^+ \setminus X_i$,

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) \leq \rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S_{\alpha,\beta}).$$

The proof is by induction on the height of a node in $\mathcal{T}$. The height of a node i in the rooted tree $\mathcal{T}$ is the distance to the farthest leaf in the subtree rooted at i. The base case is when i is a leaf node in $\mathcal{T}$ and height is 0. For a leaf node i with $X_i = \emptyset$, the row with $\alpha, \beta : \emptyset \to \{0, 1, \ldots d\}$ completes the proof of the base case. Let us assume that the claim is true for all nodes of height at most $h - 1 \geq 0$. We now prove that if the claim is true at all nodes of height at most $h - 1$, then it is true for a node of height h. Let i be a node of height h. Clearly i is not a leaf node.

When i is an introduce node. Let j be the child of i, and $X_i = X_j \cup \{v\}$ for some $v \notin X_j$. If $\beta(v) > 0$, then no feasible solution exists since $N(v) \cap (X_i^+ \setminus X_i) = \emptyset$. This is captured in our dynamic programming. In the further cases, we consider $\beta(v) = 0$. Let $L = N(v) \cap X_i$. Consider the case when $\alpha(v) = 0$. That is, the vertex v is not part of the solution. The recursive definition of the dynamic programming gives

$$S_{\alpha,\beta} = T_i[\alpha, \beta].\texttt{Solution} = T_j[\alpha', \beta'].\texttt{Solution}$$

where α' and β' are as defined in the dynamic programming. A feasible solution to the row $T_i[\alpha, \beta]$ should be feasible for the row $T_j[\alpha', \beta']$. Otherwise, the degree constraints are not met by the solution. For the solution $S = A \cup Z$,

$$\rho_d(\alpha', \beta', \mathcal{G}_j^{\alpha',\beta'}, S) \leq \rho_d(\alpha', \beta', \mathcal{G}_j^{\alpha',\beta'}, S_{\alpha,\beta})$$

since j is a node at height $h - 1$ and by our induction hypothesis. Also,

$$\rho_d(\alpha', \beta', \mathcal{G}_j^{\alpha',\beta'}, S) = \rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S)$$

since $v \notin S$. Then, we have

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) \quad = \quad \rho_d(\alpha', \beta', \mathcal{G}_j^{\alpha',\beta'}, S) \leq \rho_d(\alpha', \beta', \mathcal{G}_j^{\alpha',\beta'}, S_{\alpha,\beta}) = \rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S_{\alpha,\beta}).$$

This completes the case when $\beta(v) = 0$ and $\alpha(v) = 0$.

Now we consider the case where $\alpha(v) > 0$. In the solution $S = A \cup Z$, we need $\alpha(v)$ neighbors of U in the d-core where the set $U = (N(v) \cap X_i) \setminus (\alpha^{-1}(0) \cap \beta^{-1}(0))$. Let $S_j = S \setminus \{v\}$. There exists a $Y \subseteq U$ of size $\alpha(v)$ such that the probability $\rho_d(\alpha, \beta, G_i^{\alpha,\beta}, S)$ can be written as follows:

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) = \rho_d(\alpha_Y, \beta', \mathcal{G}_j^{\alpha_Y,\beta'}, S_j) \prod_{u \in Y} p(uv).$$

The solution S_j is compatible for the row $T_j[\alpha_Y, \beta']$ where the vertices in Y are neighbors of v and degree constraint of these vertices is decreased by 1 in α such that it will be satisfied by the edge from v. Since W in the dynamic programming is optimal over all possible $\alpha(v)$ sized subsets of W, we have

$$\begin{aligned}\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) &= \rho_d(\alpha_Y, \beta', \mathcal{G}_j^{\alpha_Y,\beta'}, S_j) \prod_{u \in Y} p(uv) \leq \rho_d(\alpha_W, \beta', \mathcal{G}_j^{\alpha_W,\beta'}, T_j[\alpha_W, \beta'].\texttt{Solution}) \prod_{u \in W} p(uv) \\ &\leq T_j[\alpha_W, \beta'].\texttt{Value} \prod_{u \in W} p(uv) = T_i[\alpha, \beta].\texttt{Value}.\end{aligned}$$

This completes the argument of the case when i is an introduce node.

When i is a forget node. Let j be the child of i, and $X_i = X_j \setminus \{v\}$ for some $v \in X_j$. We consider two cases depending on whether v belongs to S. We first consider the case when $v \notin S$. Consider the functions $\alpha_{0,\emptyset}$ and $\beta_{0,\emptyset}$ as defined in the recursive computation of forget node. Since $v \notin S$, v will get zero degree constraint in both functions α and β, and other vertices will have same constraints as α and β values. Then, the probability $\rho_d(\alpha, \beta, G_i^{\alpha,\beta}, S)$ can be written as follows:

$$\rho_d(\alpha, \beta, G_i^{\alpha,\beta}, S) = \rho_d(\alpha_{0,\emptyset}, \beta_{0,\emptyset}, G_j^{\alpha_{0,\emptyset},\beta_{0,\emptyset}}, S).$$

Since j is a node at height $h - 1$, we have

$$\begin{aligned}\rho_d(\alpha, \beta, G_i^{\alpha,\beta}, S) &= \rho_d(\alpha_{0,\emptyset}, \beta_{0,\emptyset}, G_j^{\alpha_{0,\emptyset},\beta_{0,\emptyset}}, S) \leq T_j[\alpha_{0,\emptyset}, \beta_{0,\emptyset}].\texttt{Value} \\ &\leq T_j[\alpha_t, \beta_t].\texttt{Value} = T_i[\alpha, \beta].\texttt{Value}\end{aligned}$$

Secondly, we consider the case when $v \in S$. Let $U = (N(v) \cap X_j) \setminus \alpha^{-1}(0)$. Since v is in X_j but v is not in its parent node X_i, v should have degree d in the (α, β)-constrained Probabilistic-Core problem. Then, $\alpha(v) = a$ and $\beta(v) = d - a$ for an $0 \leq a \leq d$. Let $Y \subseteq U$ of size a be the set of vertices that have an edge to v to compensate the degree constraint $\alpha(v)$ at v. Then, each vertex $u \in Y$ will gain a degree constraint $\alpha(u)$ and lose a degree constraint $\beta(u)$. Using the integer a and the set Y, the probability $\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S)$ can be written as follows:

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) = \rho_d(\alpha_{a,Y}, \beta_{a,Y}, \mathcal{G}_j^{\alpha_{a,Y},\beta_{a,Y}}, S).$$

Since j is a node at height $h - 1$, we know that the row $T_j[\alpha_{a,Y}, \beta_{a,Y}]$ is computed optimally. That is,

$$\rho_d(\alpha_{a,Y}, \beta_{a,Y}, \mathcal{G}_i^{\alpha_{a,Y},\beta_{a,Y}}, S) \leq T_j[\alpha_{a,Y}, \beta_{a,Y}].\texttt{Value} \leq T_j[\alpha_t, \beta_t].\texttt{Value} = T_i[\alpha, \beta].\texttt{Value}$$

This completes the argument of the case when i is a forget node.

When i is a join node. Let j and g be the children of i, and $X_i = X_j = X_g$. The set $S \setminus X_i$ can be partitioned into sets S_j and S_g where $S_j = (S \setminus X_i) \cap X_j$ and $S_g = (S \setminus X_i) \cap X_g$. Let $U = \{u \in X_i \mid \alpha(v) > 0\}$ and $S_i = X_i \setminus \alpha^{-1}(0)$. For each vertex $u \in U$, the degree constraint $\alpha(u)$ should be satisfied by the edges from U to u. Since $X_i = X_j = X_g$, the degree constraint $\alpha(u)$ is either satisfied in the node j or node g and not in both. Without loss of generality we assume that the degree constraint α is satisfied in the node j and no zero degree constraint α in the node g. Then we define $\alpha' : X_g \to \{0, 1, \ldots, d\}$ to be for every vertex $v \in X_i$, $\alpha'(v) = 0$. For each vertex $v \in X_i$ with $\beta(v) > 0$, the degree constraint $\beta(v)$ can be satisfied by the sets S_j and S_g together. There exists an integer $x(v)$ such that $x(v)$ neighbors in the core are obtained from the set S_j and $\beta(v) - x(v)$ neighbors in the core are obtained from the set S_g. Then, there exists a function $x : X_i \to \{0, 1, \ldots, d\}$ such that for each vertex $v \in V$, $0 \leq x(v) \leq \beta(v)$. Using the function x, the probability $\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S)$ can be given as follows:

$$\rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) = \rho_d(\alpha, \beta_x, \mathcal{G}_j^{\alpha,\beta_x}, S_i \cup S_j) \,.\, \rho_d(\alpha', \beta'_x, \mathcal{G}_g^{\alpha',\beta'_x}, S_i \cup S_g)$$

The functions β_x and β'_x for the given function x are defined in the description of dynamic programming. Since both j and g are nodes at height at most $h-1$, by induction hypothesis, the rows $T_j[\alpha_j, \beta_x]$ and $T_g[\alpha', \beta'_x]$ are computed optimally. Therefore, we have

$$\begin{aligned} \rho_d(\alpha, \beta, \mathcal{G}_i^{\alpha,\beta}, S) &= \rho_d(\alpha, \beta_x, \mathcal{G}_j^{\alpha,\beta_x}, S_i \cup S_j) \,.\, \rho_d(\alpha', \beta'_x, \mathcal{G}_g^{\alpha',\beta'_x}, S_i \cup S_g) \\ &\leq T_j[\alpha, \beta_x].\texttt{Value} \,.\, T_g[\alpha', \beta'_x].\texttt{Value} \\ &\leq T_j[\alpha, \beta_y].\texttt{Value} \,.\, T_g[\alpha', \beta'_y].\texttt{Value} = T_i[\alpha, \beta].\texttt{Value} \end{aligned}$$

where y is the optimal distribution of the degree which results in the $T_i[\alpha, \beta].\texttt{Value}$, as defined in the recursive formulation at a join node. This completes the argument for the case when i is a join node. Hence the lemma. □

For a possible world $H \sqsubseteq \mathcal{G}$, the degree of a vertex $v \in V(H)$ is at most the degree of v in $G = (V, E)$. A vertex with degree less than d in G cannot be an element of a core in any possible world. Thus, those vertices can be excluded throughout the algorithm. Such a pruning results in either the pruned graph being an empty graph or the minimum degree of the pruned graph is becoming at least d. We state the following Lemma from Koster et al. [48].

Lemma 8 (Koster et al [48])**.** *For a graph $G = (V, E)$ with $|V| \geq 2$, $\delta(G) \leq tw(G)$.*

In the following lemma, we analyze the running time of the dynamic programming algorithm.

Lemma 9. *The* PROBABILISTIC-CORE-RF *problem can be solved optimally in time $2^{\mathcal{O}(tw \log tw)} n^{\mathcal{O}(1)}$.*

Proof. For each node i in $\mathcal{T}$, our dynamic programming generates a table T_i with $(d+1)^{2tw} = \mathcal{O}(d^{2tw})$ rows. Each row $T_i[\alpha, \beta]$ for some $\alpha, \beta : X_i \to \{0, 1, \ldots, d\}$ is computed in our dynamic programming based on the type of node i. When i is leaf node, a single row exists in T_i and that is computed in $\mathcal{O}(1)$ time. When i is introduce node, we consider two cases that $\alpha(v) = 0$ and $\alpha(v) > 0$. If $\alpha(v) = 0$, then the functions α_j and β_j can be computed in time $\mathcal{O}(tw)$. If $\alpha(v) > 0$, the set U' can be found by enumerating all $\alpha(v)$ sized sets in U. This will take $\mathcal{O}(|Y|^{\beta(v)})$ time and using the upper bound on values we get $\mathcal{O}(tw^d)$. It follows that if i is an introduce node, then $T_i[\alpha, \beta]$ can be computed in time $\mathcal{O}(tw^d)$. When i is a forget node, for each value of $0 \leq a \leq min(d, |U|)$, we enumerate all a sized subsets of Y. This requires $\mathcal{O}(tw^k k)$ time. When i is join node, we need to compute the optimal distribution of $\beta(v)$ for each vertex $v \in X_i$. This requires $\mathcal{O}(d^{tw})$ time. From Lemma 8, we know that $d \leq tw$, we upper bound the value d by tw. Overall, a row $T_i[\alpha, \beta]$ can be computed in time $\mathcal{O}(tw^{tw})$. The entire table T_i can be computed in time $\mathcal{O}(tw^{3tw}) = 2^{\mathcal{O}(tw \log tw)}$. The nice tree decomposition $(X, \mathcal{T})$ has $\mathcal{O}(n \cdot tw)$ many nodes and a table on each node can be computed in time $\mathcal{O}(2^{\mathcal{O}(tw \log tw)}) n^{\mathcal{O}(1)}$. An optimal solution to the PROBABILISTIC-CORE-RF problem on the input uncertain graph $\mathcal{G}$ will be obtained from the table of the root node r. That is, $T_r[\alpha : \emptyset \to \{0\}, \beta : \emptyset \to \{0\}].\texttt{Solution}$ is the optimal solution. □

The preceding lemmas results in the following theorem.

Theorem 3. *The* PROBABILISTIC-CORE *problem admits an FPT algorithm for the parameter tw.*

6. Discussion

There are many natural questions related to the parameterized complexity of algorithms on uncertain graphs under different distribution models. The following are some open questions.

1. Are there efficient reductions between distribution models so that we can classify problems based on the efficiency of algorithms under different distribution models? This question is also

of practical significance because the distribution models are specified as sampling algorithms. Consequently, the complexity of expectation computation on uncertain graphs under different distribution models is an interesting new parameterization. Further, one concrete question is whether the LRO model is *easier* than the RF model for other optimization problems on uncertain graphs. For the two case studies considered in this paper, that is the case.

2. While our results do support the natural intuition that a tree decomposition is helpful in expectation computation, it is unclear to us how traditional techniques in parameterized algortihms can be carried over to this setting. In particular, it is unclear to us as to whether for any distribution model, a kernelization based algorithm can give an FPT algorithm on uncertain graphs.
3. We have considered the coverage and the core problems on uncertain graphs under the LRO and RF models. However, we have not been able to get an FPT algorithm with the parameter treewidth for MAX-EXP-COVER-1-RF. Actually, any approach to avoid the exponential dependence on $(\Delta\ .\ tw\ )$ would be very interesting and would give a significant insight on other approaches to evaluate the expected coverage.
4. Even though the INDIVIDUAL-CORE-RF problem and PROBABILISTIC-CORE-RF problem are similar, we have not been able to get an FPT algorithm for the INDIVIDUAL-CORE-RF problem with treewidth as the parameter. Even for other structural parameters such as vertex-cover number and feedback vertex set number, FPT results will give a significant insight on the INDIVIDUAL-CORE problem.

Author Contributions: Investigation, N.S.N. and R.V.; Writing–original draft, N.S.N. and R.V.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Añez, J.; Barra, T.D.L.; Pérez, B. Dual graph representation of transport networks. *Trans. Res. Part B Methodol.* **1996**, *30*, 209–216. [CrossRef]
2. Hua, M.; Pei, J. Probabilistic Path Queries in Road Networks: Traffic Uncertainty Aware Path Selection. In Proceedings of the 13th International Conference on Extending Database Technology (EDBT '10), Lausanne, Switzerland, 22–26 March 2010; pp. 347–358, doi:10.1145/1739041.1739084. [CrossRef]
3. Asthana, S.; King, O.D.; Gibbons, F.D.; Roth, F.P. Predicting protein complex membership using probabilistic network reliability. *Genome Res.* **2004**, *14 6*, 1170–5. [CrossRef]
4. Domingos, P.; Richardson, M. Mining the Network Value of Customers. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01), San Francisco, CA, USA, 26–29 August 2001; pp. 57–66, doi:10.1145/502512.502525. [CrossRef]
5. Frank, H. Shortest Paths in Probabilistic Graphs. *Oper. Res.* **1969**, *17*, 583–599. doi:10.1287/opre.17.4.583. [CrossRef]
6. Valiant, L.G. The Complexity of Enumeration and Reliability Problems. *SIAM J. Comput.* **1979**, *8*, 410–421. doi:10.1137/0208032. [CrossRef]
7. Hoffmann, M.; Erlebach, T.; Krizanc, D.; Mihalák, M.; Raman, R. Computing Minimum Spanning Trees with Uncertainty. In Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France, 21–23 February 2008; pp. 277–288, doi:10.4230/LIPIcs.STACS.2008.1358. [CrossRef]
8. Focke, J.; Megow, N.; Meißner, J. Minimum Spanning Tree under Explorable Uncertainty in Theory and Experiments. In Proceedings of the 16th International Symposium on Experimental Algorithms (SEA 2017), London, UK, 21–23 June 2017; pp. 22:1–22:14, doi:10.4230/LIPIcs.SEA.2017.22. [CrossRef]
9. Frank, H.; Hakimi, S. Probabilistic Flows Through a Communication Network. *IEEE Trans. Circuit Theory* **1965**, *12*, 413–414. doi:10.1109/TCT.1965.1082452. [CrossRef]
10. Evans, J.R. Maximum flow in probabilistic graphs-the discrete case. *Networks* **1976**, *6*, 161–183.10.1002/net.3230060208. [CrossRef]

11. Hassin, R.; Ravi, R.; Salman, F.S. Tractable Cases of Facility Location on a Network with a Linear Reliability Order of Links. In *Algorithms-ESA 2009, Proceedings of the 17th Annual European Symposium, Copenhagen, Denmark, 7–9 September 2009*; Springer: Berlin, Germany, 2009; pp. 275–276.
12. Hassin, R.; Ravi, R.; Salman, F.S. Multiple facility location on a network with linear reliability order of edges. *J. Comb. Optim.* **2017**, *34*, 1–25. [CrossRef]
13. Narayanaswamy, N.S.; Nasre, M.; Vijayaragunathan, R. Facility Location on Planar Graphs with Unreliable Links. In Proceedings of the Computer Science-Theory and Applications-13th International Computer Science Symposium in Russia, CSR 2018, Moscow, Russia, 6–10 June 2018; pp. 269–281. doi:10.1007/978-3-319-90530-3_23. [CrossRef]
14. Kempe, D.; Kleinberg, J.M.; Tardos, É. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 137–146, doi:10.1145/956750.956769. [CrossRef]
15. Bonchi, F.; Gullo, F.; Kaltenbrunner, A.; Volkovich, Y. Core decomposition of uncertain graphs. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), New York, NY, USA, 24–27 August 2014; pp. 1316–1325, doi:10.1145/2623330.2623655. [CrossRef]
16. Peng, Y.; Zhang, Y.; Zhang, W.; Lin, X.; Qin, L. Efficient Probabilistic K-Core Computation on Uncertain Graphs. In Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018; pp. 1192–1203, doi:10.1109/ICDE.2018.00110. [CrossRef]
17. Ball, M.O.; Provan, J.S. Calculating bounds on reachability and connectedness in stochastic networks. *Networks* **1983**, *13*, 253–278. doi:10.1002/net.3230130210. [CrossRef]
18. Zou, Z.; Li, J. Structural-Context Similarities for Uncertain Graphs. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013; pp. 1325–1330, doi:10.1109/ICDM.2013.22. [CrossRef]
19. Daskin, M.S. A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. *Transp. Sci.* **1983**, *17*, 48–70. [CrossRef]
20. Ball, M.O. Complexity of network reliability computations. *Networks* **1980**, *10*, 153–165. [CrossRef]
21. Karp, R.M.; Luby, M. Monte-Carlo algorithms for the planar multiterminal network reliability problem. *J. Complex.* **1985**, *1*, 45–64. doi:10.1016/0885-064X(85)90021-4. [CrossRef]
22. Provan, J.S.; Ball, M.O. The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM J. Comput.* **1983**, *12*, 777–788. doi:10.1137/0212053. [CrossRef]
23. Guo, H.; Jerrum, M. A Polynomial-Time Approximation Algorithm for All-Terminal Network Reliability. *SIAM J. Comput.* **2019**, *48*, 964–978. doi:10.1137/18M1201846. [CrossRef]
24. Ghosh, J.; Ngo, H.Q.; Yoon, S.; Qiao, C. On a Routing Problem Within Probabilistic Graphs and its Application to Intermittently Connected Networks. In Proceedings of the 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, Anchorage, AK, USA, 6–12 May 2007; pp. 1721–1729, doi:10.1109/INFCOM.2007.201. [CrossRef]
25. Rubino, G. *Network Performance Modeling and Simulation*; chapter Network Reliability Evaluation; Gordon and Breach Science Publishers, Inc.: Newark, NJ, USA, 1999; pp. 275–302.
26. Swamynathan, G.; Wilson, C.; Boe, B.; Almeroth, K.C.; Zhao, B.Y. Do social networks improve e-commerce?: a study on social marketplaces. In Proceedings of the first Workshop on Online Social Networks (WOSN 2008), Seattle, WA, USA, 17–22 August 2008; pp. 1–6, doi:10.1145/1397735.1397737. [CrossRef]
27. White, D.R.; Harary, F. The Cohesiveness of Blocks In Social Networks: Node Connectivity and Conditional Density. *Soc. Methodol.* **2001**, *31*, 305–359. doi:10.1111/0081-1750.00098. [CrossRef]
28. Papadimitriou, C.H.; Yannakakis, M. Shortest paths without a map. *Theor. Comput. Sci.* **1991**, *84*, 127–150. doi:10.1016/0304-3975(91)90263-2. [CrossRef]
29. Khuller, S.; Moss, A.; Naor, J. The Budgeted Maximum Coverage Problem. *Inf. Process. Lett.* **1999**, *70*, 39–45. doi:10.1016/S0020-0190(99)00031-9. [CrossRef]
30. Brown, J.J.; Reingen, P.H. Social Ties and Word-of-Mouth Referral Behavior. *J. Consum. Res.* **1987**, *14*, 350–362. [CrossRef]
31. Richardson, M.; Domingos, P.M. Mining knowledge-sharing sites for viral marketing. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 61–70, doi:10.1145/775047.775057. [CrossRef]
32. Bass, F.M. A New Product Growth for Model Consumer Durables. *Manag. Sci.* **1969**, *15*, 215–227. [CrossRef]

33. Snyder, L.V. Facility location under uncertainty: A review. *IIE Trans.* **2006**, *38*, 547–564. doi:10.1080/07408170500216480. [CrossRef]
34. Eiselt, H.A.; Gendreau, M.; Laporte, G. Location of facilities on a network subject to a single-edge failure. *Networks* **1992**, *22*, 231–246. doi:10.1002/net.3230220303. [CrossRef]
35. Colbourn, C.J.; Xue, G. A linear time algorithm for computing the most reliable source on a series-parallel graph with unreliable edges. *Theor. Comput. Sci.* **1998**, *209*, 331–345. doi:10.1016/S0304-3975(97)00124-2. [CrossRef]
36. Ding, W. Computing the Most Reliable Source on Stochastic Ring Networks. In Proceedings of the 2009 WRI World Congress on Software Engineering, Xiamen, China, 19–21 May 2009; Volume 1, pp. 345–347, doi:10.1109/WCSE.2009.31. [CrossRef]
37. Ding, W.; Xue, G. A linear time algorithm for computing a most reliable source on a tree network with faulty nodes. *Theor. Comput. Sci.* **2011**, *412*, 225–232. doi:10.1016/j.tcs.2009.08.003. [CrossRef]
38. Melachrinoudis, E.; Helander, M.E. A single facility location problem on a tree with unreliable edges. *Networks* **1996**, *27*, 219–237. doi:10.1002/(SICI)1097-0037(199605)27:3. [CrossRef]
39. Nemhauser, G.L.; Wolsey, L.A.; Fisher, M.L. An analysis of approximations for maximizing submodular set functions—I. *Math. Program.* **1978**, *14*, 265–294. doi:10.1007/BF01588971. [CrossRef]
40. Cygan, M.; Fomin, F.V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; Saurabh, S. *Parameterized Algorithms*; Springer: Berlin, Germany, 2015. doi:10.1007/978-3-319-21275-3. [CrossRef]
41. Sigal, C.E.; Pritsker, A.A.B.; Solberg, J.J. The Stochastic Shortest Route Problem. *Oper. Res.* **1980**, *28*, 1122–1129. [CrossRef]
42. Guerin, R.A.; Orda, A. QoS routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Trans. Netw.* **1999**, *7*, 350–364. doi:10.1109/90.779203. [CrossRef]
43. Günneç, D.; Salman, F.S. Assessing the reliability and the expected performance of a network under disaster risk. In Proceedings of the International Network Optimization Conference (INOC), Spa, Belgium, 22–25 April 2007.
44. Diestel, R. *Graph Theory*, 4th ed.; Graduate Texts in Mathematics; Springer: Berlin, Germany, 2012; Volume 173.
45. Bodlaender, H.L. A Tourist Guide through Treewidth. *Acta Cybern.* **1993**, *11*, 1–21.
46. Kloks, T. *Treewidth, Computations and Approximations*; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1994; Volume 842.
47. Downey, R.G.; Fellows, M.R. Fixed-parameter intractability. In Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, 22–25 June 1992.
48. Koster, A.M.C.A.; Wolle, T.; Bodlaender, H.L. Degree-Based Treewidth Lower Bounds. In Proceedings of the 4th InternationalWorkshop, WEA 2005 Experimental and Efficient Algorithms, Santorini Island, Greece, 10–13 May 2005; pp. 101–112, doi:10.1007/11427186_11. [CrossRef]