Conditional Reliability in Uncertain Graphs

Arjiit Khan, Francesco Bonchi, Francesco Gullo, and Andreas Nufer

Abstract—Network reliability is a well-studied problem that requires to measure the probability that a target node is reachable from a source node in a probabilistic (or uncertain) graph, i.e., a graph where every edge is assigned a probability of existence. Many approaches and problem variants have been considered in the literature, majority of them assuming that edge-existence probabilities are fixed. Nevertheless, in real-world graphs, edge probabilities typically depend on external conditions. In metabolic networks, a protein can be converted into another protein with some probability depending on the presence of certain enzymes. In social influence networks, the probability that a tweet of some user will be re-tweeted by her followers depends on whether the tweet contains specific hashtags. In transportation networks, the probability that a network segment will work properly or not, might depend on external conditions such as weather or time of the day.

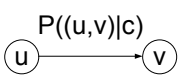
In this paper, we overcome this limitation and focus on *conditional reliability*, that is, assessing reliability when edge-existence probabilities depend on a set of conditions. In particular, we study the problem of determining the top-k conditions that maximize the reliability between two nodes. We deeply characterize our problem and show that, even employing polynomial-time reliability-estimation methods, it is $\mathbf{NP}$-hard, does not admit any $\mathbf{PTAS}$, and the underlying objective function is non-submodular. We then devise a practical method that targets both accuracy and efficiency. We also study natural generalizations of the problem with multiple source and target nodes. An extensive empirical evaluation on several large, real-life graphs demonstrates effectiveness and scalability of our methods.

Index Terms—Uncertain graphs, Reliability, Conditional probability.

1 Introduction

Uncertain graphs, i.e., graphs whose edges are assigned a probability of existence, have recently attracted a great deal of attention, due to their rich expressiveness and given that uncertainty is inherent in the data in a wide range of applications. Uncertainty may arise due to noisy measurements [2], inference and prediction models [1], or explicit manipulation, e.g., for privacy purposes [7]. A fundamental problem in uncertain graphs is the so-called *reliability*, which asks to measure the probability that two given (sets of) nodes are reachable [3]. Reliability has been well-studied in the context of device networks, i.e., networks whose nodes are electronic devices and the (physical) links between such devices have a probability of failure [3]. More recently, the attention has been shifted to other types of networks that can naturally be represented as uncertain graphs, such as social networks or biological networks [23], [31].

In the bulk of the literature, reliability queries have been modeled without taking into account any external factor that could influence the probability of existence of the links in the network. In this paper, we overcome this limitation and introduce the notion of *conditional reliability*, which takes into account that edge probabilities may depend on a set of conditions, rather being fixed. This situation models real-world uncertain graphs. As an example, Figure 1 shows a link (u, v) of a *social influence network*, i.e., a social graph where the associated probability represents the likelihood that a piece of information (e.g., a tweet) originated by u will be "adopted" (re-tweeted) by her follower v. The re-tweeting probability clearly depends on the content of the tweet. In the example, v is much more interested in sports than politics. Hence, if the tweet contains the hashtag `#NFL`, then it will likely be re-tweeted by v, while if it is about elections (i.e., it contains the hashtag `#GetToThePolls`), it will be re-tweeted only with a small probability. In this example, hashtags correspond to external factors that influence probabilities. We hereinafter refer to such external factors as *conditions* or *catalysts*, and use all these terms interchangeably throughout the paper.



c	P((u, v)|c)
#NFL	0.9
#GetToThePolls	0.2
...	...

Fig. 1: A link (u, v) of a social influence network, where the associated probability represents the likelihood that a tweet by u will be re-tweeted by her follower v. This probability depends on the content of the tweet. In this example if the tweet contains the hashtag `#NFL`, then it will likely be re-tweeted, while if it is about elections (i.e., it contains the hashtag `#GetToThePolls`), it will be re-tweeted only with a small probability.

Given an uncertain graph with external-factor-dependent edge probabilities, in this work we study the following problem: Given a source node, a target node, and a small integer k, identify a set of k catalysts that maximizes the reliability between s and t. This problem arises in many real-world scenarios, such as the ones described next.

Pathway formation in biological networks. To understand metabolic chain reactions in cellular systems, biologists utilize metabolic networks [22], where nodes represent compounds, and an edge between two compounds indicates that a compound can be transformed into another one through a chemical reaction. Reactions are controlled by various enzymes, and each enzyme defines a probability that the underlying reaction will actually take place. Thus, reactions (edges) are assigned various probabilities of existence,

- *A. Khan is with Nanyang Technological University, Singapore.*
- *F. Bonchi is with ISI Foundation, Italy.*
- *F. Gullo is with UniCredit, R&D Dept., Italy.*
- *A. Nufer is with ETH Zurich, Switzerland.*

Manuscript received January 31, 2017.

which depend on the specific enzyme (external factor). A fundamental question posed by biologists is to identify a set of enzymes which guarantee with high probability that a sequence of chemical reactions will take place to convert an input compound s into a target compound t. Since enzymes are expensive (they need to go through a long multi-step process before being commercialized [9]), the output enzyme set should be limited in size. Often known as *cost-effective experiment design* [28], [30], this corresponds to solving an instance of our problem: Given a source compound s and a target compound t, what is the set of top-k enzymes which maximizes the probability that s will be converted into t via a series of chemical reactions?

Information cascades. Studying information cascades in influential networks is receiving more and more attention, mainly due to its large applicability in *viral marketing* strategies. Social influence can be modeled as in Figure 1, i.e., by means of a probability that once u has been "activated" by a campaign, she will influence her friend v to perform the same action. This probability typically depends on topics and contents of the campaign [6], [11]. Within this view, let us consider the following example, which is motivated from [26]. During the 2016 US Presidential election, Hilary Clinton's campaign promises were infrastructure rebuild, free trade, open borders, unlimited immigration, equal pay, background checks to gun sales, increasing minimum wage, etc. To get more votes, Hilary's publicity manager could have prioritized the most influential among all these standpoints in subsequent speeches from her, her vice presidential candidate (Tim Kaine), and her political supporters (e.g., Barack and Michelle Obama), while also planning how to influence more voters from the "blue wall" states (Michigan, Pennsylvania, and Wisconsin) [32]. As speeches should be kept limited due to time constraints and risk of becoming ineffective in case of information overload, it is desirable to find a limited set of standpoints that maximize the influence from a set of early adopters (e.g., popular people who are close to Hilary Clinton) to a set of target voters (e.g., citizens of the "blue wall" states) [4]. This corresponds to identifying the top-k conditions that maximize the reliability between two (sets of) nodes in the social graph, i.e., the problem we study in this work.

Challenges and contributions. The problem that we study in this work is a non-trivial one. Computing standard reliability over uncertain graphs is a $\#\mathbf{P}$-complete problem [5]. We show that, even assuming polynomial-time sampling methods to estimate conditional reliability (such as RHT-sampling [23], recursive stratified sampling [25]), our problem of computing a set of k catalysts that maximizes conditional reliability between two nodes remains **NP**-hard. Moreover, our problem turns out to be not easy to approximate, as (i) it does not admit any **PTAS**, and (ii) the underlying objective function is shown to be non-submodular. Therefore, standard algorithms, such as iterative hill-climbing that greedily maximizes the marginal gain at every iteration, do not provide any approximation guarantees and are expected to have limited performance. Within this view, we devise a novel algorithm that first extracts highly-reliable paths between source and target nodes, and then iteratively selects these paths so as to achieve maximum improvement in reliability while still satisfying the constraint on the number of conditions.

After studying the single-source-single-target query, we focus on generalizations where multiple source and target nodes can be provided as input, thus opening the stage to a wider family of queries and applications. We study two variants of this more general problem: (i) maximizing an aggregate function over pairwise reliability between nodes in source and target sets, and (ii) maximizing the probability that source and target nodes remain all connected.

The main contributions of this paper are as follows:

- We focus on the notion of conditional reliability in uncertain graphs, which arises when the input graph has conditional edge-existence probabilities. In particular, we formulate and study the problem of finding a limited set of conditions that maximizes reliability between a source and a target node (Section 2).
- We deeply characterize our problem from a theoretical point of view, showing that it is **NP**-hard and hard to approximate even when polynomial-time reliability estimation is employed (Section 2).
- We design an algorithm that provides effective (approximated) solutions to our problem, while also looking at efficiency. The proposed method properly selects a number of highly-reliable paths so as to maximize reliability while satisfying the budget on the number of conditions (Section 4).
- We generalize our problem and algorithms to the case of multiple source and target nodes (Section 5).
- We empirically demonstrate effectiveness and efficiency of our methods on real-life graphs, while also detailing applications in information cascade (Section 6).

2 Single-source single-target: problem statement

An uncertain graph $\mathcal{G}$ is a quadruple (V, E, C, P), where V is a set of n nodes, $E \subseteq V \times V$ is a set of m directed edges, and C is a set of external conditions that influence the edge-existence probabilities. We hereinafter refer to such external conditions as *catalysts*. $P : E \times C \to (0, 1]$ is a function that assigns a conditional probability to each edge $e \in E$ given a specific catalyst $c \in C$, i.e., $P(e|c)$ denotes the probability that the edge e exists given the catalyst c.

The bulk of the literature on uncertain graphs assumes that edge probabilities are independent of one another [23]. In this work, we make the same assumption. Additionally, we assume that the existence of an edge is determined by an independent process (coin flipping), one per catalyst c, and the ultimate existence of an edge is decided based on the success of at least one of such processes. This assumption naturally holds in various settings. For instance, in a metabolic network, with an initial compound and an enzyme, the probability that a target compound would be produced depends only on that specific reaction, and it is independent of other chemical reactions defined in the network. As a result, the global existence probability of an edge e, given a set of catalysts $C_1 \subseteq C$, can be derived as $P(e|C_1) = 1 - \prod_{c \in C_1}(1 - P(e|c))$.

Given a set C_1 of catalysts, the uncertain graph $\mathcal{G}$ yields 2^m deterministic graphs $G \sqsubseteq \mathcal{G}|C_1$, where each G is a pair

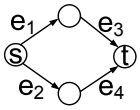


Fig. 2: Example of non-submodularity. $P(e_1|c_1) = 0.5$, $P(e_2|c_2) = 0.6$, $P(e_3|c_3) = 0.5$, $P(e_4|c_1) = 0.5$. $P(e|c) = 0$ for all other edge-catalyst combinations that are not specified.

(V, E_G), with $E_G \subseteq E$, and its probability of being observed is given below.

$$P(G|C_1) = \prod_{e \in E_G} P(e|C_1) \prod_{e \in E \setminus E_G} (1 - P(e|C_1)) \quad (1)$$

For a source node $s \in V$, and a target node $t \in V$, we define *conditional reliability* $R((s,t)|C_1)$ as the probability that t is reachable from s in $\mathcal{G}$, given a set C_1 of catalysts. Formally, for a possible graph $G \sqsubseteq \mathcal{G}|C_1$, let $I_G(s,t)$ be an indicator function taking value 1 if there exists a path from s to t in G, and 0 otherwise. $R((s,t)|C_1)$ is computed as follows.

$$R((s,t)|C_1) = \sum_{G \sqsubseteq \mathcal{G}|C_1} [I_G(s,t) \times P(G|C_1)] \quad (2)$$

The problem that we tackle in this work is introduced next.

Problem 1 (s-t TOP-k CATALYSTS)**.** *Given an uncertain graph $\mathcal{G} = (V, E, C, P)$, a source node $s \in V$, a target node $t \in V$, and a positive integer k, find a set $C^* \subseteq C$ of catalysts, having size k, that maximizes the conditional reliability $R((s,t)|C^*)$ from s to t:*

$$C^* = \arg\max_{C_1 \subseteq C} R((s,t)|C_1)$$
$$\text{subject to} \quad |C_1| = k. \quad (3)$$

Intuitively, the top-k set C^* yields multiple high-probability paths from the source node s to the target node t. Any specific path can have edges formed due to different catalysts.

Theoretical characterization. Problem 1 intrinsically relies on the classical reliability problem [1], which is $\#\mathbf{P}$-complete [5]. As a result, Problem 1 is hard as well.

However, like standard reliability, conditional reliability can be estimated in polynomial time via Monte Carlo sampling, or other sampling methods [23]. Thus, the key question is whether Problem 1 remains hard even if polynomial-time conditional-reliability estimation is employed. As formalized next, the answer to this question is positive.

Theorem 1. *Problem 1 is* $\mathbf{NP}$*-hard even assuming polynomial-time computation for conditional reliability.*

Proof. We prove $\mathbf{NP}$-hardness by a reduction from the MAX k-COVER problem. In MAX k-COVER, we are given a universe U, and a set of h subsets of U, i.e., $\mathcal{S} = \{S_1, S_2, \ldots, S_h\}$, where $S_i \subseteq U$, for all $i \in [1 \ldots h]$. The goal is to find a subset $\mathcal{S}^*$ of $\mathcal{S}$, of size $|\mathcal{S}^*| = k$, such that the number of elements covered by $\mathcal{S}^*$ is maximized, i.e., so as to maximize $|\cup_{S \in \mathcal{S}^*} S|$. Given an instance of MAX k-COVER, we construct in polynomial time an instance of s-t TOP-k CATALYSTS problem as follows.

We create an uncertain graph $\mathcal{G}$ with a source node s and a target node t. We add to $\mathcal{G}$ a set of nodes $u_1, u_2, \ldots, u_Z$, one for each element in U ($Z = |U|$). We connect each of these nodes u_i to the target node t with a (directed) edge (u_i, t), and assume that each of such edges (u_i, t) can occur only in the presence of a single catalyst c with a certain probability $p < 1$, i.e., $\forall i \in [1..Z] : P((u_i,t)|c) = p$ and $P((u_i,t)|c') = 0, \forall c' \neq c$. Similarly, we put in $\mathcal{G}$ another set of nodes $x_1, x_2, \ldots, x_Z$ (again one for each element in U), and connect each of these nodes x_i to the source node s with an edge (s, x_i). Each of such edges (s, x_i) can also be present only in the presence of catalyst c, with probability $P((s,x_i)|c) = p$. Finally, if some element $u_i \in U$ is covered by at least one of the subsets in $\mathcal{S}$, we add a directed edge (x_i, u_i) in $\mathcal{G}$. For each set $S_j \in \mathcal{S}$ that covers item u_i, we consider a corresponding catalyst c_j and set the probability $P((x_i,u_i)|c_j) = 1$.

Now, we ask for a solution of s-t TOP-k CATALYSTS on the uncertain graph $\mathcal{G}$ constructed by using $k+1$ catalysts. Every solution to our problem necessarily takes catalyst c, as otherwise there would be no way to connect s to t. Moreover, given that the paths connecting s to t are all disjoint, and each of them exists with probability < 1 (as $p < 1$), the reliability from s to t is maximized by selecting k additional catalysts that make the maximum number of paths exist, or, equivalently, selecting k other catalysts that make each of the edges (x_i, u_i) exist with probability 1. In order for each edge (x_i, u_i) to exist with probability 1, it suffices to have selected only one of the catalysts that are assigned to (x_i, u_i). Thus, selecting k catalysts that maximize the number of edges (x_i, u_i) existing with probability 1 corresponds to selecting k subsets S_j that maximize the number of elements covered. Hence, the theorem. □

Apart from being $\mathbf{NP}$-hard, Problem 1 is also not easy to approximate, as it does not admit any *Polynomial Time Approximation Scheme* ($\mathbf{PTAS}$).

Theorem 2. *Problem 1 does not admit any* $\mathbf{PTAS}$*, unless* $\mathbf{P} = \mathbf{NP}$.

Proof. See Appendix. □

As a further evidence of the difficulty of our problem, it turns out that neither submodularity nor supermodularity holds for the objective function therein. Thus, standard greedy hill-climbing algorithms do not directly come with approximation guarantees. Non-supermodularity easily follows from $\mathbf{NP}$-hardness (as maximizing supermodular set functions under a cardinality constraint is solvable in polynomial time), while non-submodularity is shown next with a counter-example.

Fact 1. *The objective function of Problem 1 is not submodular.*

A set function f is submodular if $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$, for all sets $A \subseteq B$ and all elements $x \notin B$. Look at the example in Figure 2. Let $C_1 = \{c_2\}$, $C_2 = \{c_1, c_2\}$. We find that $R((s,t)|C_1) = 0$, $R((s,t)|C_1 \cup \{c_3\}) = 0$, $R((s,t)|C_2) = 0.3$, and $R((s,t)|C_2 \cup \{c_3\}) = 0.475$. Clearly, submodularity does not hold in this example.

3 SINGLE-SOURCE SINGLE-TARGET: BASELINES

In this section, we present two simple baseline approaches and discuss their limitations (Sections 3.1 and 3.2). Then, in Section 4, we propose a more sophisticated algorithm that aims at overcoming the weaknesses of such baselines.

[1] Given an uncertain graph, a source node s, and a target node t, compute the probability that t is reachable from s.

3.1 Individual top-k baseline

The most immediate approach to our s-t TOP-k CATALYSTS problem consists of estimating the reliability $R((s,t)|\{c\})$ between the source s and the target t attained by each catalyst $c \in C$ individually, and then outputting the top-k catalysts that achieve the highest individual reliability.

Time complexity. For each catalyst, we can estimate reliability via Monte Carlo (MC) sampling[2]: sample a set of K deterministic graphs from the input uncertain graph, and estimate reliability by summing the (normalized) probabilities of the graphs where the target is reachable from the source. The time complexity of MC sampling for a single catalyst is $\mathcal{O}(K(n+m))$, where n and m denote the number of nodes and edges in the input uncertain graph, respectively. Hence, the overall time complexity of the Individual top-k baseline is $\mathcal{O}(|C|K(n+m)+|C|\log k)$, where the last term is due to top-k search.

Shortcomings. The Individual top-k algorithm suffers from both accuracy and efficiency issues.

- Accuracy: This baseline is unable to capture the contribution of paths containing different catalysts. For example, in Figure 2, the individual reliability attained by each catalyst is 0. Thus, if we are to select the top-2 catalysts, there will be no way to discriminate among catalysts, which will be picked at random. Instead, in reality, the top-2 set is $\{c_1, c_2\}$.
- Efficiency: To achieve good accuracy, MC sampling typically requires around thousands of samples [23]. Performing such a sampling for each of the $|C|$ catalysts can be quite expensive on large graphs ($|C|$ may be up to the order of thousands as well, see Section 6).

3.2 Greedy baseline

A more advanced baseline consists of greedily selecting the catalyst that brings the maximum marginal gain to the total reliability, until k catalysts have been selected. More precisely, assuming that a set C_1 of catalysts has been already computed, in the next iteration this Greedy baseline selects a catalyst c^* such that:

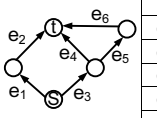
$$c^* = \arg\max_{c \in C \setminus C_1} \left[R((s,t)|C_1 \cup \{c\}) - R((s,t)|C_1)\right]$$

Note that, since the s-t TOP-k CATALYSTS problem is neither submodular nor supermodular, this greedy approach does not achieve any approximation guarantees.

Time complexity. The time complexity of each iteration of the greedy baseline is $\mathcal{O}(|C|K(n+m))$, as we need to estimate the reliability achieved by the addition of each catalyst in order to choose the one maximizing the marginal gain. For a total of k iterations (top-k catalysts are to be reported), the overall complexity is $\mathcal{O}(|C|kK(n+m))$.

Shortcomings. While being more sophisticated than Individual top-k, the Greedy baseline still suffers from both accuracy and efficiency issues.

- Accuracy: Although Greedy partially solves the accuracy issue related to the presence of paths with multiple catalysts, such an issue is still present at least in the initial phases of this second baseline.



Edge e	Catalyst c	Probability P(e|c)
e_1	c_1	0.5
e_2	c_4	0.5
e_3	c_2	0.8
e_4	c_3	0.8
e_5	c_1	0.7
e_6	c_2	0.7

Fig. 3: Difficulties with the Greedy baseline. $P(e|c) = 0$ for all edge-catalyst combinations that are not present in the table

  For example, in Figure 2 the individual reliability attained by each catalyst is 0. Therefore, in the first iteration the Greedy algorithm has no information to properly select a catalyst, thus ending up with a completely random choice. If c_3 is selected as a first catalyst, then the second catalyst selected would be c_1. Thus, Greedy would output $\{c_1, c_3\}$, while the top-2 set is $\{c_1, c_2\}$. We refer to this issue as *"cold-start"* problem.
- Efficiency: MC sampling is performed $|C|k$ times. This is more inefficient than the Individual top-k.

Example 1. *We demonstrate the cold-start problem associated with the Greedy baseline with a running example in Figure 3. Assume top-k=3. The individual reliability from s to t, attained by each of the four catalysts is 0. Therefore, in the first iteration, the Greedy algorithm selects a catalyst uniformly at random, say c_4. Then, the second catalyst selected would be c_1; since c_1, in the presence of c_4, provides the maximum marginal gain compared to any other catalyst. Similarly, in the third round, Greedy will select c_2 due to its higher marginal gain. Therefore, total reliability achieved by Greedy is: $R\big((s,t)|\{c_4,c_1,c_2\}\big) = 1-(1-0.5\times 0.5)(1-0.8\times 0.7\times 0.7) = 0.544$. However, the top-3 set is $\{c_1,c_2,c_3\}$, yielding reliability $R\big((s,t)|\{c_1,c_2,c_3\}\big) = 0.8[1-(1-0.8)(1-0.7\times 0.7)] = 0.7184$. This shows the sub-optimality of the greedy baseline.*

4 Single-source single-target: proposed method

Here we describe the method we ultimately propose to provide effective and efficient solutions to the s-t TOP-k CATALYSTS problem.

The main intuition behind our method directly follows from the shortcomings of the two baselines discussed above. Particularly, both baselines highlight how considering catalysts one at a time is less effective. This can easily be explained as a single catalyst can bring information that is related only to single edges. Instead, what really matters in computing the reliability between two nodes is the set of paths connecting the source and the target. This observation finds confirmation in the literature [12].

Motivated by this, we design the proposed method as composed of two main steps. First, we select the top-r paths exhibiting highest reliability from the source to the target. Second, we iteratively include these paths in the solution so as to maximize the marginal gain in reliability, while still keeping the constraint on total number of catalysts satisfied. Apart from the main advantage due to considering paths instead of individual catalysts, designing our algorithm as composed of two separate steps allows us to achieve high

[2]In this paper, we employ MC sampling as an oracle to estimate reliability in uncertain graphs. While more advanced sampling techniques exist, e.g., RHT [23], recursive stratified sampling [25], our contributions are orthogonal to them. We omit discussing advanced sampling methods for brevity.

Algorithm 1 Most-reliable Paths

Require: Uncertain graph $\mathcal{G} = (V, E, C, P)$, source node $s \in V$, target node $t \in V$, positive integers k, r
Ensure: Subset of catalysts $C^* \subseteq C$
1: $\mathcal{P} \leftarrow$ Algorithm 2 on input $(\mathcal{G}, s, t, r)$
2: $\mathcal{P}_1 \leftarrow$ Algorithm 3 on input $(\mathcal{G}, s, t, k)$
3: $C^* \leftarrow$catalysts present on $\mathcal{P}_1$

Algorithm 2 Top-r Most Reliable Path Selection

Require: Uncertain graph $\mathcal{G} = (V, E, C, P)$, source node $s \in V$, target node $t \in V$, positive integer r
Ensure: $\mathcal{P}$: top-r most reliable paths from s to t
1: **for all** $e \in E$ **do**
2: let $C(e) = \{c_1, c_2, \ldots, c_i\}$ be the set of all catalysts s.t. $P(e|c_j) > 0, \forall j \in [1..i]$
3: replace e by i edges $\{e_1, e_2, \ldots, e_i\}$
4: assign probability $P(e_j|c_j) = P(e|c_j)$
5: assign edge-weight $W(e_j) = -\log P(e_j|c_j)$
6: **end for**
7: $\mathcal{P} \leftarrow$ top-r shortest paths from s to t in the constructed multigraph

Algorithm 3 Iterative Path Inclusion

Require: Top-r most-reliable path set $\mathcal{P}$ from source s to target t, positive integer k
Ensure: A subset of paths $\mathcal{P}_1 \subseteq \mathcal{P}$
1: $\mathcal{P}_1 \leftarrow \emptyset$
2: **while** $|\mathcal{P}| > 0$ and total #catalysts in $\mathcal{P}_1$ is $\leq k$ **do**
3: $P^* \leftarrow \arg\max_{P \in \mathcal{P} \setminus \mathcal{P}_1} Rel_{\mathcal{P}_1 \cup \{P\}}(s, t)$ s.t. #catalysts in $\mathcal{P}_1 \cup \{P^*\}$ is $\leq k$
4: $\mathcal{P}_1 \leftarrow \mathcal{P}_1 \cup \{P^*\}$
5: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P^*\}$
6: **end while**

efficiency. Indeed, the first step can be efficiently solved by fast algorithms for finding the top-r shortest paths, while the second step requires MC sampling to be performed in a significantly reduced version of the original graph.

The outline of the proposed method, which we call Most-reliable Paths, is reported in Algorithm 1. In the following we provide the details of each of the two steps.

4.1 Step 1: Most-reliable path selection

The first step of the proposed method consists of finding the top-r most reliable paths from the source to the target. Given an uncertain graph $\mathcal{G} = (V, E, C, P)$, a source node $s \in V$, and a target node $t \in V$, we first convert $\mathcal{G}$ into an uncertain, multigraph $\mathcal{G}'$ (Algorithm 2). For each edge $e = (u, v) \in E$, let $C(e) \subseteq C$ denote the set of all *single* catalysts such that $\forall c \in C(e) : P(e|c) > 0$. Assume $C(e) = \{c_1, c_2, \ldots, c_i\}$. Then, we add i edges $\{e_1, e_2, \ldots, e_i\}$ between u and v in the multigraph $\mathcal{G}'$. To each newly constructed edge e_j, $j \in [1..i]$, we assign a single catalyst $C(e_j) = c_j$ and set $P(e_j|c_j) = P(e|c_j)$. It can be easily noted that $\mathcal{G}$ and $\mathcal{G}'$ are equivalent in terms of our problem. The construction of $\mathcal{G}'$ only serves the purpose of selecting the top-r most reliable paths from s to t in such a way that, for each intermediate pair of nodes x, y along a path, a single edge (and, thus, a single catalyst) among the many ones possibly created by the $\mathcal{G} \rightarrow \mathcal{G}'$ transformation, is picked up. The reliability of a path is defined as the product of the edge-probabilities along that path.

To ultimately compute the top-r most reliable paths, we further convert the uncertain multigraph $\mathcal{G}'$ into an edge-weighted multigraph $\mathcal{G}''$ by assigning a weight $-\log(p_e)$ to each edge e with probability p_e of $\mathcal{G}'$. This way, the top-r most reliable paths in $\mathcal{G}'$ will correspond to the top-r shortest paths in $\mathcal{G}''$. To compute the top-r shortest paths in $\mathcal{G}''$, we apply the well-established Eppstein's algorithm [16], which has time complexity $\mathcal{O}(|C|m + nlogn + r)$.

Space complexity. We note that both $\mathcal{G}'$ and $\mathcal{G}''$ can have size at most $|C|$ times more than the size of the original graph $\mathcal{G} = (V, E, C, P)$. This is because in Algorithm 2, each edge e of $\mathcal{G}$ is replaced by $C(e)$ edges in $\mathcal{G}'$ and $\mathcal{G}''$ (lines 2-3), where $C(e)$ denotes the set of all catalysts such that $\forall c \in C(e), P(e|c) > 0$. Clearly, $C(e) \leq |C|$. Therefore, both $\mathcal{G}'$ and $\mathcal{G}''$ can have at most $|E||C| = m|C|$ edges, while still having the same number of nodes as in the original graph. In summary, the size of $\mathcal{G}'$ and $\mathcal{G}''$ is *linear* in that of the original graph and in the number of catalysts. Based on our experimental results, this adds a very small overhead to the overall space requirement (see Section 6).

Choice of r. The number r of paths is an input parameter which constitutes a knob to tradeoff between efficiency and accuracy (a larger r leads to higher accuracy and lower efficiency). In general, its choice depends on the input graph. An easy yet effective way to set it is to observe when the inclusion of the top-$(r+1)$-th reliable path does not tangibly increase the reliability given by the top-r paths. We provide experimental results on selecting r in Section 6.

4.2 Step 2: Iterative path inclusion

The second step of our most-reliable-path method aims at selecting a proper subset from the top-r most-reliable path set so as to maximize reliability between source and target nodes, while also meeting the constraint on the number of output catalysts. Denoting by $Rel_{\mathcal{P}}(s, t)$ the reliability between s and t in the subgraph induced by a path set $\mathcal{P}$, this step formally corresponds to the following problem:

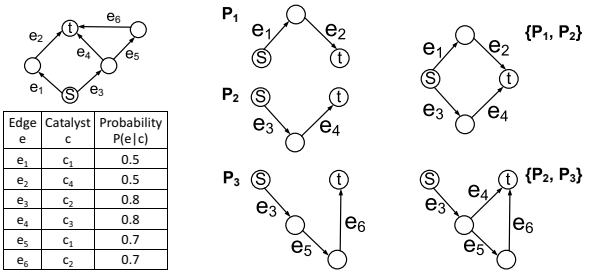
Problem 2 (ITERATIVE PATH INCLUSION)**.** *Given set $\mathcal{P}$ of top-r most reliable paths from s to t in multigraph $\mathcal{G}'$, find a path set $\mathcal{P}^* \subseteq \mathcal{P}$ such that:*

$$\mathcal{P}^* = \arg\max_{\mathcal{P}_1 \subseteq \mathcal{P}} Rel_{\mathcal{P}_1}(s, t)$$
$$\textit{subject to} \quad |\textstyle\bigcup_{e \in \mathcal{P}_1} C(e)| \leq k \tag{4}$$

The ITERATIVE PATH INCLUSION problem can be shown to be **NP**-hard via a reduction from MAX k-COVER. The proof is analogous to the one in Theorem 1, we thus omit it.

Theorem 3. *Problem 2 is* **NP***-hard.*

Algorithm. We design an efficient greedy algorithm (Algorithm 3) for the ITERATIVE PATH INCLUSION problem. At each iteration, we add a path P^* to the already computed path set $\mathcal{P}_1$ which brings the maximum marginal gain in terms of reliability. While selecting path P^*, we also ensure that the total number of catalysts used in the paths $\mathcal{P}_1 \cup \{P^*\}$ is no more than k. The algorithm terminates either when there is no path left in the top-r most reliable path set $\mathcal{P}$, or no more paths can be added without violating the budget k on the number of catalysts. We report the catalysts present in $\mathcal{P}_1$ as our final solution. If the total number of catalysts present in $\mathcal{P}_1$ is $k' < k$, additional $k - k'$ catalysts that are



Edge e	Catalyst c	Probability P(e|c)
e_1	c_1	0.5
e_2	c_4	0.5
e_3	c_2	0.8
e_4	c_3	0.8
e_5	c_1	0.7
e_6	c_2	0.7

(a) 3-paths from s to t (b) 1st Iteration (c) 2nd Iteration

Fig. 4: A demonstration of the Iterative Path Inclusion algorithm.

not in $\mathcal{P}_1$ can be selected with some proper criterion (e.g., frequency on the non-selected paths).

Next, we demonstrate our Iterative Path Inclusion algorithm with the previous running example.

Example 2. *In Figure 4(a), which is same as Figure 3, we have 3 paths from s to t, i.e., $P_1 : e_1e_2$, $P_2 : e_3e_4$, and $P_3 : e_3e_5e_6$. Assume that there is a budget constraint of 3 catalysts. In the first iteration, we select the path P_2 since it has the highest reliability compared to the two other paths. In the second iteration, P_1 and P_2 together have higher reliability than P_2 and P_3. However, the former combination requires 4 catalysts, thus violating the constraint. Hence, we select P_2 and P_3. After that, the algorithm terminates as no more path can be included without violating the constraint on catalysts.*

Approximation guarantee. The Iterative Path Inclusion algorithm achieves approximation guarantee under some assumptions. If the top-r most reliable paths are node-disjoint (except at source and target nodes), Iterative Path Inclusion exhibits an approximation ratio at least $\frac{1}{r}$.

Theorem 4. *The Iterative Path Inclusion algorithm, under the assumption that the top-r most reliable paths are node-disjoint, achieves an approximation factor of:*

$$\frac{1}{k_{Rel}}\left(1-\left(\frac{K_C-k_{Rel}}{K_C}\right)^{k_C}\right), \tag{5}$$

where

$$K_C = \max_{\mathcal{P}_1\subseteq\mathcal{P}}\{|\mathcal{P}_1| : |C(\mathcal{P}_1)| \le k\} \tag{6}$$

$$k_C = \min_{\mathcal{P}_1\subseteq\mathcal{P}}\{|\mathcal{P}_1| : |C(\mathcal{P}_1)| \le k\} \quad \textit{and}$$
$$\forall P \in \mathcal{P}\setminus\mathcal{P}_1, \quad |C(\mathcal{P}_1\cup\{P\})| > k\} \tag{7}$$

$$k_{Rel} = 1 - \min_{P\in\mathcal{P}}\frac{Rel_{\mathcal{P}}(s,t) - Rel_{\mathcal{P}\setminus\{P\}}(s,t)}{Rel_{\{P\}}(s,t)}. \tag{8}$$

Proof. See Appendix. □

In the above approximation-guarantee result, K_C and k_C, respectively, denote maximum and minimum size of the maximal feasible path set that can be derived from $\mathcal{P}$. k_{Rel} denotes the curvature of our optimization function, which can be shown to be submodular when paths in $\mathcal{P}$ are node-disjoint (see Appendix). Hence, in this case $k_{Rel} \in (0,1)$. Assuming that $\mathcal{P}$ contains at least one path having less than k catalysts, then in the worst case the approximation ratio is $\ge \frac{1}{K_C} \ge \frac{1}{r}$ (where r is the total number of paths in the top-r path set $\mathcal{P}$). In other words, the approximation ratio is guaranteed to be at least $\frac{1}{r}$.

Time complexity. Let us denote by n' and m' the number of nodes and edges, respectively, in the subgraph induced by the top-r most-reliable path set $\mathcal{P}$. At each iteration, our iterative path selection algorithm performs MC sampling over the subgraph induced by the selected paths. The number of iterations is at most r. Thus, if K is the number of samples used in each MC sampling, the iterative path selection algorithm takes $\mathcal{O}(r^2K(n'+m'))$ time. Including the time due to the first step of selecting the top-r most reliable paths, we get that the overall time complexity of the proposed Most-reliable Paths algorithm is $\mathcal{O}(|C|m + n\log n + r^2K(n'+m'))$. We point out that the subgraph induced by the top-r most reliable paths is typically much smaller than the input graph $\mathcal{G}$. Thus, our Most-reliable Paths method is expected to be much more efficient than the two baselines introduced earlier. Experiments in Section 6 confirm this claim.

5 Multiple Sources and Targets

Real-world queries often involve sets of source and/or target nodes, instead of a single source-target pair. As an example, the topic-aware information cascade problem [4] asks for a *set* of early adopters who maximally influence a given *set* of target customers. Motivated by this, in the following we discuss problems and algorithms for the case where multiple nodes can be provided as input. Such a generalization opens the stage to various formulations of the problem. Here we focus on two variants: (1) maximizing an aggregate function over all possible source-target pairs (Section 5.1), and (2) maximizing connectivity among all query nodes (Section 5.2). Note that our first problem formulation has a notion of "clique" connectivity, as it applies an aggregate function over all pairs of query nodes.

5.1 Maximizing aggregate functions

We formulate our problem as follows.

Problem 3 (top-k catalysts w/ aggregate)**.** *Given an uncertain graph $\mathcal{G} = (V, E, C, P)$, a source set $S \subset V$, a target set $T \subset V$, and a positive integer k, find a set C^* of catalysts, having size k that maximizes an aggregate function F over conditional reliability of all source-target pairs:*

$$C^* = \arg\max_{C_1\subseteq C} \underset{\langle s,t\rangle\in S\times T}{F}\big(R\left((s,t)|C_1\right)\big)$$
$$\textit{subject to} \quad |C_1| = k. \tag{9}$$

Being a generalization of the s-t TOP-k CATALYSTS problem, Problem 3 can easily be recognized as **NP**-hard. In this work we consider three commonly-used aggregate functions: average, maximum, and minimum. These aggregates give rise to three variants of Problem 3 which we refer to TOP-k CATALYSTS AVG, TOP-k CATALYSTS MAX, and TOP-k CATALYSTS MIN, respectively:

- **Average.** Find the top-k catalysts such that the average reliability over all $\langle s,t\rangle$ pairs is maximized. This is equivalent to maximization of the sum of reliability over all $\langle s,t\rangle$ pairs. This problem occurs, e.g., in the topic-aware information cascade scenario when the campaigner wants to maximize the spread of information to the entire target group.

- **Maximum.** Find the top-k catalysts such that the reliability of the $\langle s,t\rangle$ pair with the highest reliability is maximized. In the topic-aware information cascade problem, this is equivalent to the scenario that each early adopter is campaigning a different product of the same campaigner. The campaigner wants at least one target user to be aware about one of her products (e.g., each target user might be a celebrity user in Twitter). Therefore, the campaigner would be willing to maximize the spread of information from at least one early adopter to at least one target user.
- **Minimum.** Find the top-k catalysts such that the reliability of the $\langle s,t\rangle$ pair having the lowest reliability is maximized. In the topic-aware information cascade setting, this is equivalent to the problem that each early adopter is campaigning a different product of the same campaigner, and the campaigner wants to maximize the minimum spread of her campaign from any of the early adopters to any of her target users. This is motivated, in reality, because only a small percentage of the users who have heard about a campaign will buy the corresponding product.

Overview of algorithms. In the following, we describe the algorithms that we develop for the aforementioned aggregate functions. In principle, they follow our earlier two main steps: (1) finding the top-r paths (Algorithm 2), now between every pair of source and target nodes, and then (2) iteratively include these paths so as to maximize the marginal gain in regards to the objective function, while still keeping the constraint on total number of catalysts satisfied (Algorithm 3). However, the exact process somewhat varies according to the problem at hand, which we shall discuss next. Unless otherwise specified, we assume that $S \cap T = \emptyset$, that is, source and target sets are non-overlapping. We will discuss case by case how our algorithms can (easily) handle the case when $S \cap T \neq \emptyset$.

Extending the baselines presented in Sections 3.1–3.2 to multiple query nodes is instead straightforward (regardless of the aggregate function). We thus omit the details.

5.1.1 Algorithm for maximum reliability

Our solution for the TOP-k CATALYSTS MAX problem is the most straightforward, compared to both TOP-k CATALYSTS AVG and TOP-k CATALYSTS MIN problems. First, for each $\langle s,t\rangle$ pair, we identify the top-r most reliable paths. Then, separately for each $\langle s,t\rangle$ pair, we also apply the Iterative Path Inclusion algorithm to find the top-k catalysts for that pair. Finally, we select the $\langle s,t\rangle$ pair which attains the maximum reliability. We report the corresponding top-k catalysts as the solution to the TOP-k CATALYSTS MAX problem.

Time complexity. The time required to find the reliable paths for all $\langle s,t\rangle$ pairs is $\mathcal{O}\left(|S||T|\left(m+n\log n+r\right)\right)$. Similarly, the time complexity of the iterative path inclusion phase is $\mathcal{O}\left(|S||T|r^2\left(n'+m'\right)K\right)$. There is an additional cost $\mathcal{O}(|S||T|)$ to find the $\langle s,t\rangle$ pair with the maximum reliability, which is, however, dominated by the time spent in path inclusion.

Overlapping source and target sets. If a node v is both in S and in T, the above solution will return an arbitrary set C_1 of catalysts, since $R\left((v,v)|C_1\right)=1$, and it will always be considered as the optimal solution. If this behavior is not intended, we eliminate all such nodes in $S \cap T$ before applying our algorithm.

5.1.2 Algorithm for average reliability

As earlier, we first identify the top-r most reliable paths for each $\langle s,t\rangle$ pair. However, we are now interested in the *average* reliability considering all source and target nodes, as opposed to that for individual source-target pairs. Thus, we consider *all* selected $|S||T|r$ paths together, and apply the Iterative Path Inclusion algorithm to add paths so to maximize the marginal gain in terms of our objective function, while maintaining the budget k on total number of catalysts. Recall that here our objective function is $\frac{1}{|S||T|}\sum_{\langle s,t\rangle\in S\times T}\left(R\left((s,t)|C_1\right)\right)$. Finally, catalysts present in the selected paths are reported as a solution to the TOP-k CATALYSTS AVG problem.

The above steps remain identical, regardless of whether S and T overlap or not.

Time complexity. The time required to find the reliable paths for all $\langle s,t\rangle$ pairs is $\mathcal{O}\left(|S||T|\left(m+n\log n+r\right)\right)$, as we apply Eppstein's algorithm for $|S||T|$ times. Next, the time complexity of the iterative path inclusion phase is $\mathcal{O}\left(\left(|S||T|r\right)^2\left(n'+m'\right)K\right)$, where n' and m' are the number of nodes and edges of the subgraph induced by the reliable paths, and K is the number of samples used in each MC sampling. Note that the time required for iterative path inclusion of TOP-k CATALYSTS AVG is higher than that for the TOP-k CATALYSTS MAX, since we consider all $|S||T|r$ paths together in the former algorithm.

5.1.3 Algorithm for minimum reliability

We start again by finding the top-r most reliable paths for each $\langle s,t\rangle$ pair. However, applying the Iterative Path Inclusion algorithm, in this case, is more subtle. Specifically, if there are many $\langle s,t\rangle$ pairs and a limited budget k of catalysts, spending too many catalysts for a single $\langle s,t\rangle$ pair might prevent us from finding a path for another pair. This way there will be pairs with conditional reliability very low, thus the solution to the TOP-k CATALYSTS MIN problem, i.e., the pair exhibiting minimum conditional reliability, would be quite poor. To mitigate this issue, we consider an additional step where we find a minimum set of catalysts, before applying the Iterative Path Inclusion algorithm. The subsequent steps remain instead identical, regardless of whether S and T overlap or not.

Finding minimum set of catalysts. The objective of this step is to select a minimum set of catalysts which ensure that at least one path for every $\langle s,t\rangle$ pair exists. This step corresponds to the following problem.

Problem 4 (MINIMUM SET OF CATALYSTS)**.** *Given a source set S, a target set T, a set of paths $\mathcal{P}$, an uncertain graph $\mathcal{G}=(V,E,C,P)$ induced by $\mathcal{P}$, find the smallest set $C^*\subseteq C$ of catalysts, such that the conditional reliability $R\left((s,t)|C^*\right)$ for each $\langle s,t\rangle$ pair is larger than zero:*

$$C^* = \arg\min_{C_1\subseteq C}|C_1|$$
$$\text{subject to} \quad R\left((s,t)|C_1\right)>0,\ \forall\langle s,t\rangle\in S\times T. \tag{10}$$

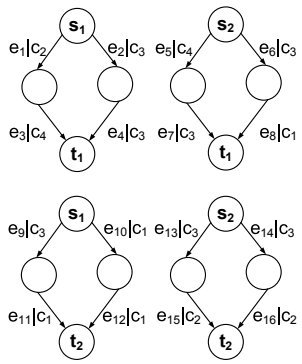


Fig. 5: A demonstration of TOP-k CATALYSTS MIN solution: catalysts that influence the edges are shown in the figure. However, the corresponding probabilities are not shown. Also, $P(e|c) = 0$ for all edge-catalyst combinations that are not present in the figure.

Theorem 5. *Problem 4 is* **NP***-hard.*

Proof. **NP**-hardness can easily be verified by noticing that the SET COVER problem can be reduced to a specific instance of MINIMUM SET OF CATALYSTS where each path in $\mathcal{P}$ can be formed using a single catalyst. In this case, in fact, we ask for the minimum number of catalysts required to cover at least one path of every $\langle s, t\rangle$ pair, which exactly corresponds to what SET COVER asks for. □

Algorithm for MINIMUM SET OF CATALYSTS. We design an algorithm to provide effective solutions to MINIMUM SET OF CATALYSTS which consists of four steps:

- **Step 1.** Mark all $\langle s, t\rangle$ pairs as disconnected.
- **Step 2.** For all disconnected $\langle s, t\rangle$ pairs, find a path P that connects one of such $\langle s, t\rangle$ pairs, while adding the minimum number of new catalysts to the set of already selected catalysts.
- **Step 3.** Mark that $\langle s, t\rangle$ pair as connected. Include the catalysts in path P to the set of selected catalysts.
- **Step 4.** If there is at least one disconnected $\langle s, t\rangle$ pair, go to step 2.

We report the set of selected catalysts as our minimum set. If the size of this minimum set is more than k, we perform an additional step. From the selected set C^*, if a subset C' can be removed, but a path can still be found for all connected $\langle s, t\rangle$ pairs with the remaining catalysts in $C^* \setminus C'$, then C' is removed from C^*. We illustrate our algorithm with an example below.

Example 3. *As shown in Figure 5, let us assume that the source set is $S = \{s_1, s_2\}$, and the target set is $T = \{t_1, t_2\}$. The figure illustrates the top-2 most reliable paths for each source-target pair. Assume there is a budget $k = 3$ on the number of output catalysts. We now apply our algorithm. First, we select the catalyst c_3, since this catalyst is sufficient to have an edge for the $\langle s_1, t_1\rangle$ pair. Then, we select the catalyst c_4 because $\{c_3, c_4\}$ together add an edge for the $\langle s_2, t_1\rangle$ pair. Next, we consider the catalyst c_1 in order to have an edge for the pair $\langle s_1, t_2\rangle$. At this point, we have already saturated the budget of 3 catalysts: $\{c_1, c_3, c_4\}$, but we are yet to add an edge for the $\langle s_2, t_2\rangle$ pair. Thus, we delete c_4 from the selected set of catalysts, because this still allows a path for three previously connected source-target pairs. Finally, we add catalyst c_2 to the set. The final set $\{c_1, c_2, c_3\}$ of catalysts allows a path for all source-target pairs.*

Time complexity. In each iteration, we find the path with the smallest number of new catalysts, which requires $\mathcal{O}\left(r|S||T|\left(n' + m'\right)\right)$ time. Then, we also remove the redundant catalysts, which requires another $\mathcal{O}\left(kr|S||T|\left(n' + m'\right)\right)$ time. Since there can be at most $|S||T|$ iterations, overall time complexity of our MINIMUM SET OF CATALYSTS finding algorithm is $\mathcal{O}\left(kr|S|^2|T|^2\left(n' + m'\right)\right)$.

Intuitively, the minimum set finding step ensures that, given large enough budget on catalysts, there will be at least one path for all $\langle s, t\rangle$ pairs. Therefore, the objective function of the TOP-k CATALYSTS MIN problem will be guaranteed to be larger than zero. If our budget has not been exhausted yet and more catalysts can be added, we next apply the Iterative Path Inclusion algorithm as follows. At each iteration, we find the $\langle s, t\rangle$ pair exhibiting the minimum conditional reliability. We then add a path that maintains the budget, while also maximizing the marginal gain in reliability for that $\langle s, t\rangle$ pair. The algorithm terminates when no more paths can be added without exceeding the budget k, or all top-r paths for all $\langle s, t\rangle$ pairs have been selected.

5.2 Maximizing connectivity

In the second variant of the s-t TOP-k CATALYSTS problem applied to multiple query nodes, we do not distinguish between source and target nodes. All query nodes are considered as peers: the objective of this CONNECTIVITY TOP-k CATALYSTS problem is to find a set of top-k catalysts which maximize the probability that all query nodes are connected in the subgraph induced by edges containing those catalysts. An application of this problem is finding a suitable topic list of a thematic scientific event among researchers. The event would be successful not only when the invitees are experts on those topics, but also if they can network with each other, that is, they can find connections (e.g., direct and indirect links formed due to research collaborations) with other invitees based on those topics [33]. We formally define our problem below.

Problem 5 (CONNECTIVITY TOP-k CATALYSTS)**.** *Given an uncertain graph $\mathcal{G} = (V, E, C, P)$, a set of query nodes $Q \subset V$, and a positive integer k, find a set C^* of catalysts, with size k, that maximizes the probability of nodes in Q being connected only using catalysts in C^*:*

$$C^* = \arg\max_{C_1 \subseteq C} \sum_{G \sqsubseteq \mathcal{G}|C_1} \left[J_G(Q) \times P(G|C_1)\right]$$
$$\text{such that} \quad |C_1| = k. \tag{11}$$

In the above statement, $J_G(Q)$ is an indicator function over a possible deterministic graph $G \sqsubseteq \mathcal{G}|C_1$ taking value 1 if nodes in Q are all connected in G, and 0 otherwise. For simplicity, in directed graphs we consider a weak notion of connectivity, i.e., connectivity disregarding edge-directions. The extension to strong connectivity is straightforward.

Algorithm. The CONNECTIVITY TOP-k CATALYSTS problem is a generalization of the s-t TOP-k CATALYSTS basic problem (Problem 1). Thus, it can be immediately be recognized as **NP**-hard.

To provide high quality results, we design a two-step algorithm whose outline is similar in spirit to the Most-reliable Paths algorithm proposed for

dataset	nodes	edges	catalysts	edge probabilities: mean, SD, quartiles
DBLP	1 291 297	3 561 816	347	0.21, 0.08, {0.181, 0.181, 0.181}
BioMine	1 045 414	6 742 943	20	0.27, 0.17, {0.116, 0.216, 0.363}
Freebase	28 483 132	46 708 421	5 428	0.50, 0.24, {0.250, 0.500, 0.750}

TABLE 1: Characteristics of the uncertain graphs used in the experiments.

s-t TOP-k CATALYSTS. As a main difference, however, since our goal in CONNECTIVITY TOP-k CATALYSTS is to maximize *connectivity* among a set of peer nodes, we ask for the top-r minimum *Steiner trees* as a first step of the algorithm (rather than top-r most reliable paths between a single source-target pair). A Steiner tree for a set Q of nodes in a weighted graph is a tree that spans all nodes of Q. A minimum Steiner tree is a Steiner tree whose sum of edge-weights is the minimum. We first apply the technique proposed in [15] to find the top-r minimum Steiner trees from an equivalent edge-weighted, multi-graph $\mathcal{G}''$. We recall that $\mathcal{G}''$ can be obtained from the input uncertain graph $\mathcal{G}$ by following Algorithm 2. Next, we iteratively include the Steiner trees in our solution so as to maximize the marginal gain in the probability that nodes in Q are connected, while not exceeding the budget on catalysts.

Time complexity. The complexity to find the top-r minimum Steiner trees is $\mathcal{O}\left(3^{|Q|}n + 2^{|Q|}\left((|Q| + \log n)\, n + e\right)\right)$ [15]. As for Iterative Path Inclusion, the complexity of our iterative tree inclusion method is $\mathcal{O}(r^2(n' + m')K)$, where, we recall, K is the number of samples used in each MC sampling, and n' and m' are the number of nodes and edges in the subgraph induced by the top-r minimum Steiner trees, respectively.

6 Experimental evaluation

We report empirical results to show accuracy, efficiency, and memory usage of the proposed methods. We also provide results on information diffusion to demonstrate the applicability of the top-k catalysts identified by our methods. We report sensitivity analysis by varying all main parameters: the number of catalysts, reliable paths, query nodes, and the distance between source and target nodes. The code is implemented in C++ and experiments are performed on a single core of a 100GB, 2.26GHz Xeon server.

6.1 Experimental setup

Datasets. We use three real-world uncertain graphs.

DBLP (http://dblp.uni-trier.de/xml). We use this well-known collaboration network, downloaded on August 31, 2016. Each node represents an author, and an edge denotes co-authorship. Each edge is defined by a set of keywords, that are present within the title of the papers, co-authored by the respective authors. We selected 347 distinct keywords from all paper titles, e.g., databases, distributed, learning, crowd, verification, etc, based on frequency and how well they represent various sub-areas of computer science. We count occurrences of a specific keyword in the titles of the papers co-authored by any two authors. Edge probabilities are derived from an exponential cdf of mean $\mu = 5$ to this count [23]; hence, if a keyword c appeared t times in the titles of the papers co-authored by the authors u and v, the corresponding probability is $p((u, v)|c)) = 1 - exp^{-t/5}$. The intuition is that the more the times u and v co-authored on keyword c, the higher the chance (i.e., the probability) that u influences v (and, vice versa) for that keyword. Therefore, keywords correspond to catalysts for information cascade.

BioMine (https://www.cs.helsinki.fi/group/biomine). This is the database of the BIOMINE project [17]. The graph is constructed by integrating cross-references from several biological databases. Nodes represent biological concepts such as genes, proteins, etc., and edges denote real-world phenomena between two nodes, e.g., a gene "codes" for a protein. In our setting these phenomena correspond to catalysts. Edge probabilities, which quantify the existence of a phenomenon between the two endpoints of that edge, were determined in [17] as a combination of three criteria: relevance (i.e., relative importance of that relationship type), informativeness (e.g., degrees of the nodes adjacent to that edge), confidence on the existence of a specific relationship (e.g., conformity with the biological STRING database).

Freebase (http://www. freebase.com). This is a knowledge graph, where nodes are named entities (e.g., Google) or abstract concepts (e.g., Asian people), while edges represent relationships among those entities (Jerry Yang is the "founder" of Yahoo!). Relationships corresponds to catalysts. We use the probabilistic version of the graph as derived in [10].

Query selection. For each set of experiments, we select 500 different queries. If we do not impose any distance constraint between the source and the target, both of them are picked uniformly at random. When we would like to maintain a maximum pairwise distance d from the source to the target, we first select the source uniformly at random. Then, out of all nodes that are within d-hops from it, one node is selected uniformly at random as the target. All reported results are averaged over 500 such queries.

Competing methods. We compare the proposed Most-reliable Paths method (Algorithm 1) to the two baselines, Individual top-k and Greedy, discussed in Sections 3.1–3.2. For the sake of brevity, in the remainder of this section we refer to the proposed method as Rel-Path, and to the Individual top-k baseline as Ind-k.

Reliability estimation. Our proposed method and the baselines need a subroutine that estimates conditional reliability for given source node(s), target node(s), and number of catalysts. To this end, we employ the well-established Monte Carlo-sampling method. In particular, to improve efficiency, we combine MC sampling with a breadth first search from the source node (set) [23], meaning that the coin for establishing if an edge should be included in the current sample is flipped only upon request. This avoids to flip coins for edges in parts of the graph that are not reached with the current breadth first search, thus increasing the chance of an early termination. In the experiments, we found that MC sampling converges at around $K = 1000$ samples in our datasets. This is roughly the same number observed in the literature [23], [31] for these datasets. Hence, we set $K = 1000$ in all sets of experiments.

6.2 Single-source single-target

Experiments over different datasets. In Table 2, we show conditional reliability and running time of all competitors for top-5 output catalysts. For our Rel-Path, we use top-20 most reliable paths with *Freebase* and *BioMine* and top-50 most reliable paths over *DBLP*, as we observe that, for finding the top-5 catalysts, increasing the number of paths beyond 20 (*Freebase* and *BioMine*) and 50 (*DBLP*) does

	conditional reliability			running time (sec)		
dataset	Ind-k	Greedy	Rel-Path	Ind-k	Greedy	Rel-Path
Freebase	0.15	0.15	**0.17**	1.38	43	**0.02**
BioMine	0.18	0.43	**0.59**	1220	26217	**5.27**
DBLP	0.11	0.26	**0.28**	85.97	36519	**1.07**

TABLE 2: Reliability and efficiency over different datasets. Single source-target pair, top-5 catalysts.

	conditional reliability			running time (sec)		
k	Ind-k	Greedy	Rel-Path	Ind-k	Greedy	Rel-Path
5	0.18	0.43	**0.59**	1220	26217	**5.27**
8	0.18	0.49	**0.59**	2210	67158	**7.05**
10	0.18	0.50	**0.60**	2290	131674	**7.37**
12	0.23	0.53	**0.62**	2305	161265	**7.98**
15	0.34	0.53	**0.63**	2365	217496	**8.30**

TABLE 3: Reliability and efficiency with varying number k of output catalysts. Single source-target pair, *BioMine* dataset.

distance	conditional reliability			running time (sec)		
(# hops)	Ind-k	Greedy	Rel-Path	Ind-k	Greedy	Rel-Path
2	0.45	0.75	**0.83**	346	9798	**4.90**
4	0.08	0.38	**0.64**	406	23140	**5.37**
6	0.02	0.17	**0.30**	548	29135	**5.58**

TABLE 4: Reliability and efficiency with varying distance between the source and the target. Single source-target pair, *BioMine* dataset, top-5 catalysts.

not significantly increase the quality in respective datasets. Results with varying the number of most reliable paths, and its dependence on varying number of top-k catalysts, will be reported shortly.

Conditional reliability illustrates the quality of the top-k catalysts found: the higher the reliability, the better the quality. The proposed Rel-Path achieves the best quality results on all our datasets.

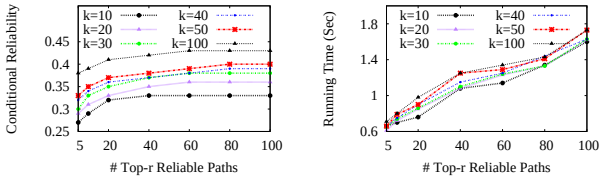
Concerning running time, we observe that Rel-Path is 2-3 orders of magnitude faster than Ind-k, and 3-4 orders faster than Greedy. This confirms that performing MC sampling on a significantly reduced version of the input graph leads to significant benefits in terms of efficiency, without affecting accuracy. Surprisingly, Greedy is orders of magnitude slower than Ind-k. The reason is the following. Although only a factor k separates Ind-k from Greedy based on our complexity analysis, what happens in practice is that Ind-k benefits from MC-sampling's early termination much more than Greedy, as Ind-k considers each catalyst individually, while Greedy considers a set of catalysts. One may also note that the running times over *BioMine* is higher than that over *Freebase*. Although *Freebase* has more nodes and edges, the graph is sparse compared to *BioMine*. Therefore, a breadth first search in *BioMine* often traverses more nodes, thus increasing its processing time.

Varying number of catalysts. We show results with varying the number k of output catalysts in Table 3 and Figure 6. Similar trends have been observed in all datasets, thus, for brevity, we report results on *BioMine* (Table 3, k varies from 5 to 15) and on *DBLP* (Figure 6, k varies from 10 to 100). As expected, conditional reliability and running time increase with more catalysts. Moreover, as shown in Table 3, our Rel-Path remains more accurate and faster than both baselines for all k.

Varying distance from the source to the target. Table 4 reports on results with varying the distance between the source and the target. Keeping fixed the number of output catalysts, as expected, the reliability achieved by all three methods decreases with larger distance from the source to the target. However, we observe that the reliability drops sharply for Ind-k. This is because with increasing distance, it becomes less likely that there would be a path due to only one catalyst from the source to the target. We also note that the reliability decreases more in Greedy than in the proposed Rel-Path. This is due to the cold-start problem of Greedy: It is more likely for Greedy to make mistaken choices in the initial steps if the source and the target are connected by longer paths.

r	conditional reliability Rel-Path	running time (sec) Rel-Path
1	0.27	4.29
2	0.29	4.26
3	0.31	4.30
4	0.31	4.30
5	0.31	4.31
10	0.32	4.31
15	0.32	4.37
20	0.33	5.26
30	0.33	5.29
50	0.33	5.38
100	0.33	5.70

TABLE 5: Reliability and efficiency with varying number r of most reliable paths in the proposed Rel-Path. Single source-target pair, top-5 catalysts, *BioMine*.



(a) Conditional Reliability (b) Running Time

Fig. 6: Reliability and efficiency with varying number r of most reliable paths in the proposed Rel-Path. Single source-target pair, number of top-k catalysts vary from k=10 to k=100, *DBLP*.

Datasets	Memory Usage
DBLP (1.3M, 3.6M)	1.9 GB
BioMine (1.0M, 6.7M)	1.8 GB
Freebase (28.5M, 46.7M)	16.0 GB

TABLE 6: Memory usage for Rel-Path

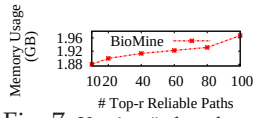
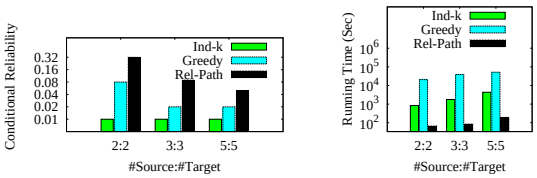


Fig. 7: Varying #rel. paths

Varying number of most reliable paths. We also test Rel-Path for different values of the number r of most reliable paths discovered in the first step of the algorithm. We report these results for *BioMine* in Table 5, and for *DBLP* in Figure 6. For the *BioMine* dataset, we fix the number k of output catalysts as 5, and we observe that while increasing the number of paths, the reliability initially increases, then saturates at a certain value of r (e.g., $r = 20$). This behavior is expected, since the subsequent paths have very small reliability. Hence, including them does not significantly increase the quality of the solution found so far. On the other hand, the running time increases almost linearly when more top-r paths are considered.

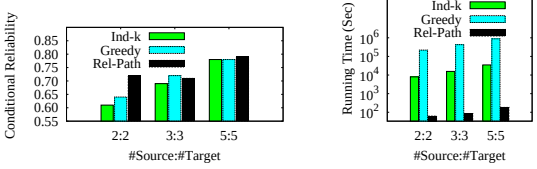
A similar behavior is observed in the *DBLP* dataset. Here we additionally vary the number k of output catalysts from 10 to 100 (Figure 6), and we find that, as k increases, a larger set of reliable paths need to be considered to make accuracy stabilize. For instance, for $k = 10$, about $r = 20$ reliable paths suffice to observe no more tangible accuracy improvement. On the other hand, for $k = 100$, $r = 60$ paths are required. Once again, this behavior is expected: the larger the number k of catalysts to be output, the larger the subgraph connecting source to target to be explored, and, hence, the larger the number of paths to be considered so as to satisfactorily cover that subgraph.

Memory usage. We report the memory usage of Rel-Path in Table 6. This is dominated by the space required for the graph in the main memory. The top-r reliable paths selected by our algorithm consumes only a few tens of megabytes.



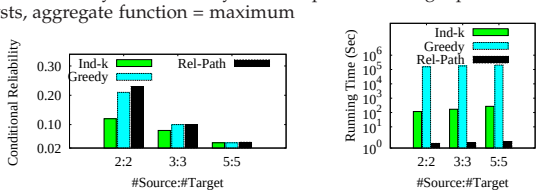
(a) Conditional Reliability (b) Running Time

Fig. 8: Reliability and efficiency for multiple source-target pairs: *Freebase*, top-5 catalysts, aggregate function = minimum.



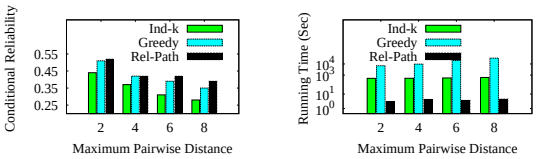
(a) Reliability (b) Running Time

Fig. 9: Reliability and efficiency for multiple source-target pairs: *Freebase*, top-5 catalysts, aggregate function = maximum



(a) Reliability (b) Running Time

Fig. 10: Reliability and efficiency for multiple source-target pairs: *DBLP*, top-10 catalysts, aggregate function = average



(a) Reliability (b) Running Time

Fig. 11: Reliability and efficiency for multiple source-target pairs with varying distances: *BioMine*, top-5 catalysts, agg. func. = average, #Source=3, #Target=3

Moreover, the memory consumption increases linearly with the number of reliable paths selected (Figure 7).

6.3 Multiple-sources multiple-targets

Aggregate functions. We perform experiments to evaluate the reliability and efficiency of our methods that maximize an aggregate function over conditional reliabilities for many source-target pairs. We consider Minimum aggregate function, and vary the number of source and target nodes from 2 to 5. In these experiments, we fix the maximum distance between any source-target pair as 2. We also ensure that the same node is not included both in source and target sets.

We show the performance of our algorithms over *Freebase* (Figure 8). Similar to queries with single source-target pairs, Rel-Path outperforms Ind-k and Greedy both in terms of efficiency and conditional reliability. Particularly, due to the presence of multiple source-target pairs, running time differences scale up, and Rel-Path is at least four orders of magnitude faster than the baselines.

We find that with more source-target pairs, the minimum reliability achieved decreases (Figures 8(a)). This can be explained as follows. As we keep the number of top-k catalysts fixed at $k = 5$, with more source and target nodes, the likelihood of getting one source-target pair with small reliability attained by those top-k catalysts increases.

Different aggregate functions and datasets. We demonstrate how our aggregate functions perform over *Freebase* and *DBLP*, in Figures 9 and 10, respectively. Due to common trends, we only show Maximum over *Freebase* and Average over *DBLP*. We find that Rel-Path results in better reliability compared to Greedy over all experiments. Their difference minimizes in both datasets with more source-target pairs, which is due to the fact that we keep the number of top-k catalysts fixed at $k = 5$ (for *Freebase*) and at $k = 10$ (for *DBLP*). As before, Rel-Path is at least four to five orders of magnitude faster than Greedy in all scenarios. In particular, Greedy requires about 10^5 seconds to answer a single query, which makes almost infeasible to apply this baseline technique in any real-world online application.

Datasets	Connectivity			Running Time (Sec)		
	Ind-k	Greedy	Rel-Path	Ind-k	Greedy	Rel-Path
Freebase	0.01	**0.10**	**0.10**	1 908	13 175	**80**
BioMine	0.29	0.47	**0.71**	893	37 992	**310**
DBLP	0.30	0.33	**0.35**	306	116 340	**85**

TABLE 7: Connectivity query with 4 nodes, top-5 catalysts

It is interesting that with more source and target pairs, the maximum reliability increases (Figure 9), but the average reliability decreases (Figure 10). This is expected since with more source-target pairs, the chance of getting one pair with higher reliability also increases, thereby improving the maximum reliability. On the contrary, as we consider more source-target pairs while keeping the total number of catalysts same, the average reliability naturally decreases.

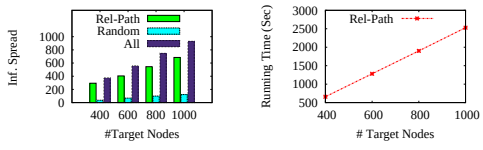
Varying distance from source set to target set. We vary the distance between source and target nodes as follows. We first select one node uniformly at random in the graph, and then select our source and target sets from the h-hop neighborhood of that node. By considering $h = 1, 2, 3,$ and 4, we ensure that the maximum pairwise distance between any source and target is bounded by 2, 4, 6, and 8-hops, respectively. In Figure 11, we show our reliability and efficiency results over *BioMine* and with Avergae aggregate function. With larger distance, the reliability achieved by all three methods decreases, and the running time increases. However, even with maximum pairwise distance 8, Rel-Path is five orders of magnitude faster than Greedy.

Connectivity maximization. We illustrate the performance of our algorithms that maximize connectivity (defined in Section 5.2) across multiple query nodes. For these experiments, we select 4 query nodes with maximum pairwise distance between any two nodes fixed at 2. We compare the connectivity attained by top-5 catalysts in Table 7. It can be observed that Greedy and Rel-Path perform equally well in *Freebase*, whereas Rel-Path results in higher connectivity over *BioMine* and *DBLP*. We further analyze the top-20 Steiner trees retrieved in BioMine, and find that each of these Steiner trees require 3$\sim$5 distinct catalysts. Therefore, in this dataset, Greedy makes more mistakes at initial stages. Because of the complexity of the top-20 Steiner tree finding algorithm, Rel-Path requires more running time in these experiments. However, Rel-Path is still significantly faster that the other two baselines over all our datasets.

6.4 Application in information cascade

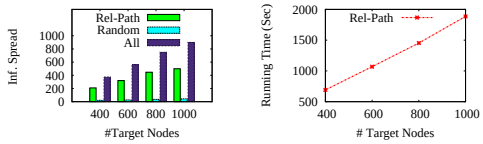
Here we showcase our top-k catalysts problem in the context of information diffusion over social networks. We present our results over the *DBLP* dataset.

We select top-k catalysts (i.e., keywords) according to the **Average** aggregate function for multiple sources and targets



(a) Information Spread (b) Running Time

Fig. 12: (a) Expected information spread by the top-10 catalysts and (b) running time to find the top-10 catalysts: *DBLP*, DB source nodes, DB target nodes



(a) Information Spread (b) Running Time

Fig. 13: (a) Expected information spread by the top-10 catalysts and (b) running time to find the top-10 catalysts: *DBLP*, ARCH source nodes, DB target nodes

(see Section 5). As discussed earlier, if u and v co-authored more on keyword c, the higher is the chance (i.e., the probability) that u influences v (and, vice versa) for that keyword. Therefore, keywords correspond to catalysts for information cascade. *The ultimate goal of this application is to show that the catalysts selected by our method effectively accomplish the task of maximizing the expected spread of information between the source nodes and the target nodes*. To this purpose, we measure the expected spread achieved by the top-k catalysts selected by our method, and compare it to the expected spread achieved by (*i*) k random catalysts, and (*ii*) *all* catalysts.

Source nodes from Databases. We find the top-10 authors having the maximum number of publications in top-tier database conferences and journals. They are: {Divesh Srivastava, Surajit Chaudhuri, Jiawei Han, Philip S. Yu, Hector Garcia-Molina, Jeffrey F. Naughton, H. V. Jagadish, Michael Stonebraker, Beng Chin Ooi, Raghu Ramakrishnan}.

Source nodes from Computer Architecture. In an analogous manner, we select the top-10 authors from the computer architecture domain: {Alberto L. Sangiovanni-Vincentelli, Jingsheng Jason Cong, Massoud Pedram, Andrew B. Kahng, Robert K. Brayton, Yao-Wen Chang, David Blaauw, Miodrag Potkonjak, Kaushik Roy, Xianlong Hong}.

Target nodes from Databases. We consider authors having at least 5 publications in top-tier database conferences and journals as our target nodes. We vary the number of target nodes from 400 to 1000, selected uniformly at random from them, to demonstrate the scalability of our algorithm.

In Figures 12(a) and 13(a), we show the expected information spread achieved by the top-10 catalysts selected via our Rel-Path method, under two scenarios, respectively, **Case-1:** both source nodes and target nodes are from databases (DB), **Case-2:** source nodes are from architecture (ARCH), and target nodes from databases (DB). To demonstrate the quality of our results, we report the expected information spread achieved by uniformly at random selection of 10 catalysts (*denoted as "*Random*" in the figures*). We observe from Figures 12(a) and 13(a) that Rel-Path selects high-quality catalysts, and significantly outperforms such a Random method.

In particular, the catalysts selected by Rel-Path under Case-1 are all DB-related, e.g., database systems, relational, information extraction, keyword search, XML, data mining, etc. On the other hand, the catalysts selected by Rel-Path under Case-2 belong to DB or ARCH areas, e.g., CMOS, FPGA, storage system, cache, VLSI circuit, On-chip, transactional memory, data stream, etc. Both in Figures 12(a) and 13(a), we also report the total information spread achieved by all 5 428 catalysts (i.e., keywords) present in the *DBLP* dataset. This is denoted as "All" in the figures. We find that *the information spread achieved by only the top-10 catalysts is generally within 70-90% of the total information spread achieved by all 5 428 catalysts*. These results demonstrate the relevance of our novel problem and its solution in the domain of information cascade over social influence networks.

Furthermore, we find that that running time to find the top-k catalysts via Rel-Path increases almost linearly with more target nodes (see Figures 12(b) and 13(b)), which illustrates the *scalability* of our technique.

7 Related Work

To the best of our knowledge, the problem of finding the top-k catalysts for maximizing the conditional reliability, that we study in this work, is novel. In the following, we provide an overview of relevant work in neighboring areas.

Reliability queries in uncertain graphs. Reliability is a classic problem studied in systems and device networks [3]. Reliability has been recently studied in the context of large social and biological networks. Due to its #**P**-completeness [5], efficient sampling, pruning, and indexing methods have been considered [23], [25], [31].

Constrained reachability queries. Mendelson and Wood show that finding all simple paths in a (deterministic) graph matching a regular expression is **NP**-hard [29]. There are some query languages which support regular expression queries only in some restricted form, e.g., GraphQL, SoQL, GLEEN, XPATH, and SPARQL. Fan et. al. [19] study a special case of regular expressions that can be solved in quadratic time. Edge-label constrained reachability and distance queries have been studied in [8], [22].

Label-constrained reachability queries have been also considered in the context of uncertain graphs [10]. However, in that work the goal was to estimate the reliability between two nodes under the constraint that paths connecting the two nodes contain only some admissible labels. Thus, the input graph still has fixed edge probabilities that do not vary based on external conditions. As a result, label-constrained reachability differs from conditional reliability introduced in this work, and, more importantly, our problem of finding the top-k external conditions is not addressed in those works.

Explaining relationships among entities. Several works aim at identifying the best subgraphs/paths to describe how some input entities are related [18], [20], [33]. Sun et. al. propose PathSIM [34] to find entities that are connected by similar relationship patterns. However, all these works consider deterministic graphs. The semantics behind the notion of connectivity in uncertain graphs is different.

Uncertain graphs with correlated edge probabilities. Although the bulk of the literature on uncertain graphs assumes edges to be independent of one another [7], [10], [23], [25], some works deal with correlated edge probabilities, where the existence of an edge may depend on the existence of other edges in the graph (typically, edges sharing an end node) [13], [14], [31], [35]. Another model that differs from the classic one is the one adopted by Liu

et. al. [27], which considers that every edge is assigned a (discrete) probability density function over a set of possible edge weights. However, none of those works model edge-existence probabilities conditioned on external factors, nor they study the problem of finding the top-k factors that maximize the reliability between two (sets of) nodes.

Topic-aware influence maximization. The classical problem of influence maximization has been recently considered in a topic-aware fashion [6], [11]. Although the input to that problem is similar to the input considered in this work (an uncertain graph where edge probabilities depend on some conditions), topic-aware influence maximization solves a different problem, i.e., finding a set of seed nodes that maximize the spread of information for a given topic set. Topic-aware influence maximization can however benefit from the solutions provided by our top-k catalysts problem, e.g., in the case where topics are not known in advance. A recent work by Li et. al. [26] focuses on the problem of finding a size-k tag set that maximizes the *expected spread* of influence started from a given source node. Our work is different as we aim at finding the top-k external factors maximizing the *reliability* between two given (sets of) nodes.

Difference with our prior work. A preliminary version of this work was published as a short paper in [24]. The present version contains a lot of new significant material: a complete piece of research work concerning the generalization to the case of multiple source/target nodes, including problem formulations, theory, applications, algorithms, and experiments; important theoretical findings; more details, examples, and motivations for all the proposed algorithms, including detailed time-complexity analyses; a lot of additional experiments, including applications in information cascade; a detailed overview of the related literature.

8 Conclusions

We formulated and investigated a novel problem of identifying the top-k catalysts that maximize the reliability between source and target nodes in an uncertain graph. We proposed a method based on iterative reliable-path inclusion. Our experiments show that the proposed method achieves better quality and significantly higher efficiency compared to simpler baselines. In future, we shall consider more complex relationships between an edge and the catalysts, and other problems from the perspective of top-k catalysts, e.g., nearest neighbors and influence maximization.

9 Acknowledgement

Arijit Khan is supported by MOE Tier-1 M401020000 and NTU M4081678. Any opinions, findings, and conclusions in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] E. Adar and C. Re. Managing Uncertainty in Social Networks. *IEEE Data Eng. Bull.*, 2007.
[2] C. Aggarwal. *Managing and Mining Uncertain Data*. Springer, 2009.
[3] K. K. Aggarwal, K. B. Misra, and J. S. Gupta. Reliability Evaluation A Comparative Study of Different Techniques. *Micro. Rel.*, 14(1), 1975.
[4] S. Aral and D. Walker. Creating Social Contagion Through Viral Product Design: A Randomized Trial of Peer Influence in Networks. *Management Science*, 57(9):1623–1639, 2011.
[5] M. O. Ball. Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Tran. on Reliability*, 1986.
[6] N. Barbieri, F. Bonchi, and G. Manco. Topic-Aware Social Influence Propagation Models. In *ICDM*, 2012.
[7] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting Uncertainty in Graphs for Identity Obfuscation. *PVLDB*, 5(11):1376–1387, 2012.
[8] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen. Distance Oracles in Edge-Labeled Graphs. In *EDBT*, 2014.
[9] M. Chaplin and C. Bucke. *Enzyme Technology*. Cambridge University Press, 1990.
[10] M. Chen, Y. Gu, Y. Bao, and G. Yu. Label and Distance-Constraint Reachability Queries in Uncertain Graphs. In *DASFAA*, 2014.
[11] S. Chen, J. Fan, G. Li, J. Feng, K. L. Tan, and J. Tang. Online Topic-aware Influence Maximization. *PVLDB*, 8(6):666–677, 2015.
[12] W. Chen, C. Wang, and Y. Wang. Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. In *KDD*, 2010.
[13] Y. Cheng, Y. Yuan, G. Wang, B. Qiao, and Z. Wang. Efficient Sampling Methods for Shortest Path Query over Uncertain Graphs. In *DASFAA*, 2014.
[14] Y.-R. Cheng, Y. Yuan, L. Chen, and G.-R. Wang. Threshold-Based Shortest Path Query over Large Correlated Uncertain Graphs. *JCST*, 30(4):762–780, 2015.
[15] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding Top-k Min-Cost Connected Trees in Databases. In *ICDE*, 2007.
[16] D. Eppstein. Finding the k Shortest Paths. *SIAM J. Comput*, 28(2):652–673, 1998.
[17] L. Eronen and H. Toivonen. Biomine: Predicting Links between Biological Entities using Network Models of Heterogeneous Databases. *BMC Bioinformatics*, 13(1), 2012.
[18] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast Discovery of Connection Subgraphs. In *KDD*, 2004.
[19] W. Fan, J. Li, S. Ma, N. Tang, and Y. Wu. Adding Regular Expressions to Graph Reachability and Pattern Queries. In *ICDE*, 2011.
[20] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. REX: Explaining Relationships Between Entity Pairs. *PVLDB*, 5(3):241–252, 2011.
[21] R. K. Iyer and J. A. Bilmes. Submodular Optimization with Submodular Cover and Submodular Knapsack Constraints. In *NIPS*, 2013.
[22] R. Jin, H. Hong, H. Wang, N. Ruan, and Y. Xiang. Computing Label-Constraint Reachability in Graph Databases. In *SIGMOD*, 2010.
[23] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-Constraint Reachability Computation in Uncertain Graphs. *PVLDB*, 4(9):551–562, 2011.
[24] A. Khan, F. Gullo, T. Wohler, and F. Bonchi. Top-k Reliable Edge Colors in Uncertain Graphs. In *CIKM*, 2015.
[25] R. Li, J. X. Yu, R. Mao, and T. Jin. Efficient and Accurate Query Evaluation on Uncertain Graphs via Recursive Stratified Sampling. In *ICDE*, 2014.
[26] Y. Li, J. Fan, D. Zhang, and K.-L. Tan. Discovering Your Selling Points: Personalized Social Influential Tags Exploration. In *SIGMOD*, 2017.
[27] Z. Liu, C. Wang, and J. Wang. Aggregate Nearest Neighbor Queries in Uncertain Graphs. In *WWW*, 2014.
[28] J.-L. Ma, B.-C. Yin, X. Wu, and B.-C. Ye. Simple and Cost-Effective Glucose Detection Based on Carbon Nanodots Supported on Silver Nanoparticles. *Analytical Chemistry*, 89(2):1323–1328, 2017.
[29] A. O. Mendelzon and P. T. Wood. Finding Regular Simple Paths in Graph Databases. *SIAM J. Comput.*, 24(6):1235–1258, 1995.
[30] E. F. Murphy, S. G. Gilmour, and M. J. C. Crabbe. Efficient and Cost-Effective Experimental Determination of Kinetic Constants and Data: The Success of a Bayesian Systematic Approach to Drug Transport, Receptor Binding, Continuous Culture and Cell Transport Kinetics. *FEBS Letters*, 556(1):193 – 198, 2004.
[31] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-Nearest Neighbors in Uncertain Graphs. *PVLDB*, 3(1-2):997–1008, 2010.
[32] M. Sieff. Why Hillary Clinton Lost Her Blue Wall. http://www.martinsieff.com/ cycles-of-change/ hillary-clinton-lost-blue-wall/, 2016.
[33] M. Sozio and A. Gionis. The Community-search Problem and How to Plan a Successful Cocktail Party. In *KDD*, 2010.
[34] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *PVLDB*, 4(11):992–1003, 2011.
[35] Y. Yuan, G. Wang, H. Wang, and L. Chen. Efficient Subgraph Search over Large Uncertain Graphs. *PVLDB*, 4(11):876–886, 2011.

APPENDIX

Proof of Theorem 2. A problem is said to admit a *Polynomial Time Approximation Scheme* (**PTAS**) if the problem admits a polynomial-time constant-factor approximation algorithm for *every* constant $\beta \in (0,1)$. We prove the theorem by showing that there exists at least one value of β such that, if a β-approximation algorithm for s-t TOP-k CATALYSTS exists, then we can solve the well-known SET COVER problem in polynomial time. Since SET COVER is an **NP**-hard problem, clearly this can happen only if **P** = **NP**.

In SET COVER we are given a universe U, and a set of h subsets of U, i.e., $\mathcal{S} = \{S_1, S_2, \ldots, S_h\}$, where $S_i \subseteq U$, for all $i \in [1 \ldots h]$. The decision version of SET COVER asks the following question: given k, is there any a solution with no more than k sets that cover the whole universe?

Given an instance of SET COVER, we construct in polynomial time an instance of our s-t TOP-k CATALYSTS problem in the same way as in Theorem 1. On this instance, if k sets suffice to cover the whole universe in the original instance of SET COVER, the optimal solution C^* would have reliability at most $[1-(1-p^2)^Z]$, where $Z = |U|$ (because at most Z disjoint paths from s to t would be produced, each with existence probability p^2). On the other hand, if no k sets cover the whole universe, C^* would have reliability at most $[1-(1-p^2)^{Z-1}]$ (because at least one of the disjoint paths would be discarded).

Now, assume that a polynomial-time β-approximation algorithm for s-t TOP-k CATALYSTS exists, for some $\beta \in (0,1)$. Call it "Approx". Approx would yield a solution C_2 such that $R((s,t)|C_2) \geq \beta R((s,t)|C^*)$. Now, consider the inequality $[1-(1-p^2)^{Z-1}] < \beta[1-(1-p^2)^Z]$. If this inequality has solution for some values of β and p, then by simply running Approx on the instance of s-t TOP-k CATALYSTS constructed this way, and checking the reliability of the solution returned by Approx, one can answer SET COVER in polynomial time: a solution to SET COVER exists iff the solution given by Approx has reliability $\geq \beta[1-(1-p^2)^Z]$. Thus, to prove the theorem we need to show that a solution to that inequality exists.

To this end, consider the real-valued function $f(p,Z) = \frac{1-(1-p^2)^{Z-1}}{1-(1-p^2)^Z}$. Our inequality has a solution iff $\beta > f(p,Z)$. It is easy to see that $f(p,Z) < 1$, for all $Z \geq 1$ and $p > 0$. This means that there will always be a value of $\beta \in (0,1)$ and p for which $\beta > f(p,Z)$ is satisfied, regardless of Z. Hence, there exists at least one value of β such that the inequality $[1-(1-p^2)^{Z-1}] < \beta[1-(1-p^2)^Z]$ has solution, and, based on the above argument, such that no β-approximation algorithm for Problem 1 can exist. The theorem follows.

Proof of Theorem 4. If both our objective function and the constraint were proved to be submodular, our iterative path inclusion problem (Problem 2) would become an instance of the *Sub-modular Cost Sub-modular Knapsack* (SCSK) problem [21], and the approximation result in Theorem 4 would easily follow from [21]. In the following we show that indeed both our objective function (Lemma 1) and our constraints (Lemma 2) are submodular, thus also proving Theorem 4.

Lemma 1. *The constraint of the iterative path inclusion problem, i.e., total number of catalysts on edges of the included paths is sub-modular with respect to inclusion of paths.*

Proof. Consider two path sets $\mathcal{P}_1$, $\mathcal{P}_2$ from s to t such that $\mathcal{P}_2 \supseteq \mathcal{P}_1$. Also, we assume a path P from s to t, where $P \notin \mathcal{P}_2$. There can be two distinct cases: (a) P has no common catalyst with the paths in $\mathcal{P}_2 \setminus \mathcal{P}_1$. (b) P has at least one common catalyst with the paths in $\mathcal{P}_2 \setminus \mathcal{P}_1$. In the first case,

$$\left|\bigcup_{e\in\mathcal{P}_1\cup\{P\}} C(e)\right| - \left|\bigcup_{e\in\mathcal{P}_1} C(e)\right| = \left|\bigcup_{e\in\mathcal{P}_2\cup\{P\}} C(e)\right| - \left|\bigcup_{e\in\mathcal{P}_2} C(e)\right| \quad (12)$$

In the second case,

$$\left|\bigcup_{e\in\mathcal{P}_1\cup\{P\}} C(e)\right| - \left|\bigcup_{e\in\mathcal{P}_1} C(e)\right| < \left|\bigcup_{e\in\mathcal{P}_2\cup\{P\}} C(e)\right| - \left|\bigcup_{e\in\mathcal{P}_2} C(e)\right| \quad (13)$$

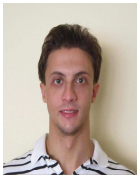
Hence, the result follows. □

Lemma 2. *If the top-r most reliable paths are node-disjoint (except at source and target nodes), then the objective function of the iterative path selection problem (Problem 2), i.e., $Rel_{\mathcal{P}_1}(s,t)$ is sub-modular with respect to inclusion of paths.*

Proof. Assume $\mathcal{P}_1, \mathcal{P}_2 \subset \mathcal{P}$, such that $\mathcal{P}_1 \subseteq \mathcal{P}_2$. Also consider a path $P \in \mathcal{P}$ and $P \notin \mathcal{P}_2$. Let us denote by $Rel_{\mathcal{P}_1}(s,t) = p_1$, $Rel_{\mathcal{P}_1\cup\{P\}}(s,t) = p_1 + \delta$, and $Rel_{\mathcal{P}_2\setminus\mathcal{P}_1}(s,t) = p_2$. Due to our assumption that the top-r most reliable paths in $\mathcal{P}$ are node-disjoint except at the source and the target, we have: $Rel_{\mathcal{P}_2}(s,t) = 1-(1-p_1)(1-p_2)$, and $Rel_{\mathcal{P}_2\cup\{P\}}(s,t) = 1-(1-p_1-\delta)(1-p_2)$. Hence, $Rel_{\mathcal{P}_2\cup\{P\}}(s,t) - Rel_{\mathcal{P}_2}(s,t) = (1-p_2)\delta$. This is smaller than or equal to δ, which was the marginal gain for including the path P in the set $\mathcal{P}_1$. Therefore, our objective function is sub-modular. □

Arijit Khan is an Assistant Professor in the School of Computer Science and Engineering at Nanyang Technological University, Singapore. He earned his PhD from the University of California, Santa Barbara, and did a post-doc in the Systems group at ETH Zurich. Arijit is the recipient of the IBM PhD Fellowship in 2012-13. He co-presented tutorials on graph queries and systems at ICDE 2012, VLDB 2014, 2015, 2017.

Francesco Bonchi is a Research Leader at the ISI Foundation, Turin, Italy. Before he was Director of Research at Yahoo Labs in Barcelona, Spain. He is an Associate Editor of the IEEE Transactions on Knowledge and Data Engineering (TKDE), and the ACM Transactions on Intelligent Systems and Technology (TIST). Dr. Bonchi has served as program co-chair of HT 2017, ICDM 2016, and ECML PKDD 2010. He earned his Ph.D. from the University of Pisa in 2003.



Francesco Gullo is a research scientist at UniCredit,, R&D department. He received his Ph.D. from the University of Calabria (Italy) in 2010. He spent four years at Yahoo Labs, Barcelona, first as a postdoctoral researcher, and, starting from 2013, as a research scientist. He served as workshop chair of ICDM'16, and organized several workshops/symposia (MIDAS @ECML-PKDD'16, MultiClust @SDM'14, @KDD'13).

Andreas Nufer is a Senior Consultant at GridSoft AG in Switzerland. He completed his masters in Computer Science at ETH Zurich, and did his bachelors at Ecole Superieure en Sciences Informatiques in France, and also from Bern University of Applied Science in Switzerland. He worked as a software developer in Distalogic GmbH and in Swisscom AG.