RESULTS reproduced

Clustering Uncertain Graphs

Matteo Ceccarello
Department of Information Engineering
University of Padova (Italy)
ceccarel@dei.unipd.it

Carlo Fantozzi
Department of Information Engineering
University of Padova (Italy)
carlo.fantozzi@unipd.it

Andrea Pietracaprina
Department of Information Engineering
University of Padova (Italy)
capri@dei.unipd.it

Geppino Pucci
Department of Information Engineering
University of Padova (Italy)
geppo@dei.unipd.it

Fabio Vandin
Department of Information Engineering
University of Padova (Italy)
vandinfa@dei.unipd.it

ABSTRACT

An uncertain graph $\mathcal{G} = (V, E, p : E \to (0, 1])$ can be viewed as a probability space whose outcomes (referred to as *possible worlds*) are subgraphs of $\mathcal{G}$ where any edge $e \in E$ occurs with probability $p(e)$, independently of the other edges. These graphs naturally arise in many application domains where data management systems are required to cope with uncertainty in interrelated data, such as computational biology, social network analysis, network reliability, and privacy enforcement, among the others. For this reason, it is important to devise fundamental querying and mining primitives for uncertain graphs. This paper contributes to this endeavor with the development of novel strategies for clustering uncertain graphs. Specifically, given an uncertain graph $\mathcal{G}$ and an integer k, we aim at partitioning its nodes into k clusters, each featuring a distinguished center node, so to maximize the minimum/average connection probability of any node to its cluster's center, in a random possible world. We assess the NP-hardness of maximizing the minimum connection probability, even in the presence of an oracle for the connection probabilities, and develop efficient approximation algorithms for both problems and some useful variants. Unlike previous works in the literature, our algorithms feature provable approximation guarantees and are capable to keep the granularity of the returned clustering under control. Our theoretical findings are complemented with several experiments that compare our algorithms against some relevant competitors, with respect to both running-time and quality of the returned clusterings.

PVLDB Reference Format:
Matteo Ceccarello, Carlo Fantozzi, Andrea Pietracaprina, Geppino Pucci, Fabio Vandin. Clustering Uncertain Graphs. *PVLDB*, 4(11): 472 - 484, 2017.
DOI: 10.1145/3164135.3164143

Proceedings of the VLDB Endowment, Vol. 4, No. 11

DOI: 10.1145/3164135.3164143

1. INTRODUCTION

In the big data era, data management systems are often required to cope with uncertainty [2]. Also, many application domains increasingly produce interrelated data, where uncertainty may concern the intensity or the confidence of the relations between individual data objects. In these cases, graphs provide a natural representation for the data, with the uncertainty modeled by associating an existence probability to each edge. For example, in Protein-Protein Interaction (PPI) networks, an edge between two proteins corresponds to an interaction that is observed through a noisy experiment characterized by some level of uncertainty, which can thus be conveniently cast as the probability of existence of that edge [3]. Also, in social networks, the probability of existence of an edge between two individuals may be used to model the likelihood of an interaction between the two individuals, or the influence of one of the two over the other [22, 1]. Other applications of uncertainty in graphs arise in the realm of mobile ad-hoc networks [6, 16], knowledge bases [8], and graph obfuscation for privacy enforcement [7]. This variety of application scenarios calls for the development of fundamental querying and mining primitives for uncertain graphs which, as argued later, can become computationally challenging even for graphs of moderate size.

Following the mainstream literature [30], an *uncertain graph* $\mathcal{G} = (V, E, p)$ is defined over a set of nodes V, a set of edges E between nodes of V, and a probability function $p : E \to (0, 1]$. $\mathcal{G}$ can be viewed as a probability space whose outcomes (referred to as *possible worlds*, in accordance with the terminology adopted for probabilistic databases [5, 13]) are graphs $G = (V, E')$ where any edge $e \in E$ is included in E' with probability $p(e)$, independently of the other edges. The main objective of this work is to introduce novel strategies for clustering uncertain graphs, aiming at partitioning the node set V so to maximize two connectivity-related objective functions, which can be seen as reinterpretations of the objective functions of the classical k-center and k-median problems [31] in the framework of uncertain graphs.

In the next subsection we provide a brief account on the literature on uncertain graphs most relevant to our work.

1.1 Related work

Early work on network reliability has dealt implicitly with the concept of uncertain graph. In general, given an uncer-

tain graph, if we interpret edge probabilities as the complement of failure probabilities, a typical objective of network reliability analysis is to determine the probability that a given set of nodes is connected under random failures. This probability can be estimated through a Monte Carlo approach, which however becomes prohibitively cumbersome for very low reliability values. In fact, even the simplest problem of computing the exact probability that two distinguished nodes s and t are connected is known to be $\#P$-complete [4, 33]. In the last three decades, several works have tried to come up with better heuristics for various reliability problems on uncertain graphs (see [21] and references therein). Some works have studied various formulations of the problem of determining the most reliable source in a network subject to edge failures, which are special cases of the clustering problems studied in this paper (see [14] and references therein).

The definition of uncertain graph adopted in this paper has been introduced in [30], where the authors investigate various probabilistic notions of distance between nodes, and develop efficient algorithms for determining the k nearest neighbors of a given source under their different distance measures. It has to be remarked that the proposed measures do not satisfy the triangle inequality, thus ruling out the applicability of traditional metric clustering approaches. In the last few years, there has been a multitude of works studying several analytic and mining problems on uncertain graphs. A detailed account of the state of the art on the subject can be found in [28], where the authors also investigate the problem of extracting a representative possible world providing a good summary of an uncertain graph for the purposes of query processing.

A number of recent works have studied different ways of clustering uncertain graphs, which is the focus of this paper. In [23] the authors consider, as a clustering problem, the identification of a deterministic *cluster graph*, which corresponds to a clique-cover of the nodes of the uncertain graph, aiming at minimizing the expected *edit distance* between the clique-cover and a random possible world of the uncertain graph, where the edit distance is measured in terms of edge additions and deletions. A 5-approximation algorithm for this problem is provided in [23]. The main drawback of this approach is that the formulation of the clustering problem does not allow to control the number of clusters. Moreover, the approximate solution returned by the proposed algorithm relies on a shallow star-decomposition of the topology of the uncertain graph, which always yields a large number of clusters (at least $|V|/(\Delta+1)$, where Δ is the maximum degree of a node in V). Thus, the returned clustering may not exploit more global information about the connectivity properties of the underlying topology.

The same clustering problem considered in [23] has been also studied by Gu et al. in [18] for a more general class of uncertain graphs, where the assumption of edge independence is lifted and the existence of an edge (u, v) is correlated to the existence of its adjacent edges (i.e., edges incident on either u or v). The authors propose two algorithms for this problem, one that, as in [23], does not fix a bound on the number of returned clusters, and another that fixes such a bound. Neither algorithm is shown to provide worst-case guarantees on the approximation ratio.

In [25] a clustering problem is defined with the objective of minimizing the expected entropy of the returned clustering, defined with respect to the adherence of the clustering to the connected components of a random possible world. With this objective in mind, the authors develop a clustering algorithm which combines a standard k-means strategy with the Monte Carlo sampling approach for reliability estimation. No theoretical guarantee is offered on the quality of the returned clustering with respect to the defined objective function, and the complexity of the approach, which does not appear to scale well with the size of the graph, also depends on a convergence parameter which cannot be estimated analytically. In summary, while the pursued approach to clustering has merit, there is no rigorous analysis of the tradeoffs that can be exercised between the quality of the returned clustering and the running time of the algorithm.

In [34] the *Markov Cluster Algorithm* (MCL) is proposed for clustering weighted graphs. In MCL, an edge weight is considered as a *similarity score* between the endpoints. The algorithm does not specifically target uncertain graphs, but it can be run on these graphs by considering the edge probabilities as weights. In fact, some of the aforementioned works on the clustering of uncertain graphs have used MCL for comparison purposes. The algorithm focuses on finding so-called *natural clusters*, that is, sets of nodes characterized by the presence of many edges and paths between their members. The basic idea of the algorithm is to perform random walks on the graph and to partition the nodes into clusters according to the probability of a random walk to stay within a given cluster. Edge weights (i.e., the similarity scores) are used by the algorithm to define the probability that a given random walk traverses a given edge. The algorithm's behavior is governed by a parameter, called *inflation*, which indirectly controls the granularity of the clustering. However, there is no fixed analytic relation between the inflation parameter and the number of returned clusters, since the impact of the inflation parameter is heavily dependent on the graph's topology and on the edge weights. The author of the algorithm maintains an optimized and very efficient implementation of MCL, against which we will compare our algorithms in Section 5.

Finally, it is worth mentioning that the problem of influence maximization in a social network under the Independent Cascade model, introduced in [22], can be reformulated as the search of k nodes that maximize the expected number of nodes reachable from them on an uncertain graph associated with the social network, where the probability on an edge (u, v) represents the likelihood of u influencing v. A constant approximation algorithm for this problem, based on a computationally heavy Monte Carlo sampling, has been developed in [22], and a number of subsequent works have targeted faster approximation algorithms (see [9, 32] and references therein). It is not clear whether the solution of this problem can be employed to partition the nodes into k clusters that provide good approximations for our two objective functions.

1.2 Our Contribution

In this paper we develop novel strategies for clustering uncertain graphs. As observed in [25], a good clustering should aim at high probabilities of connectivity within clusters, which is however a hard goal to pursue, both because of the inherent difficulty of clustering per se, and because of the aforementioned #P-completeness of reliability estima-

tion in the specific uncertain graph scenario. Also, it has been observed [30, 25] that the straightforward reduction to shortest-path based clustering where edge probabilities become weights may not yield significant outcomes because it disregards the possible world semantics.

Motivated by the above scenario, we will adopt the *connection probability* between two nodes (a.k.a. two-terminal reliability), that is, the probability that the two nodes belong to the same connected component in a random possible world, as the distance measure upon which we will base our clustering. As a first technical contribution, which may be of independent interest for the broader area of network reliability, we show that this measure satisfies a form of triangle inequality, unlike other distance measures used in previous works. This property allows us to cast the problem of clustering uncertain graphs into the same framework as traditional clustering approaches on metric spaces, while still enabling an effective integration with the possible world semantics is encapsulated in the use of the connection probability.

Specifically, we study two clustering problems, together with some variations. Given in input an n-node uncertain graph $\mathcal{G}$ and an integer k, we seek to partition the nodes of $\mathcal{G}$ into k clusters, where each cluster contains a distinguished node, called *center*. We will devise approximation algorithms for each of the following two optimization problems: (a) maximize the Minimum Connection Probability of a node to its cluster center (MCP problem); and (b) maximize the Average Connection Probability of a node to its cluster center (ACP problem).

We first prove that the MCP problem is NP-hard even in the presence of an oracle for the connection probability, and make the plausible conjecture that the ACP problem is NP-hard as well. Our approximation algorithms for the MCP and ACP problems are both based on a simple deterministic strategy that computes a *partial* k-clustering aiming at covering a maximal subset of nodes, given a threshold on the minimum connection probability of a node to its cluster's center. By incorporating this strategy within suitable guessing schedules, we are able to obtain k-clusterings with the following guarantees:

- for the MCP problem, minimum connection probability $\Omega\left(p^2_{\mathrm{opt-min}}(k)\right)$, where $p_{\mathrm{opt-min}}(k)$ is the maximum minimum connection probability of any k-clustering.
- for the ACP problem, average connection probability $\Omega\left(\left(p_{\mathrm{opt-avg}}(k)/\log n\right)^3\right)$, where $p_{\mathrm{opt-avg}}(k)$ is the maximum average connection probability of any k-clustering.

We also discuss variants of our algorithms that allow to impose a limit on the length of the paths that contribute to the connection probability between two nodes. Computing provably good clusterings under limited path length may have interesting applications in scenarios such as the analysis of PPI networks, where topological distance between two nodes diminishes their similarity, regardless of their connection probability.

We first present our clustering algorithms assuming the availability of an oracle for the connection probabilities between pairs of nodes. Then, we show how to integrate a progressive sampling scheme for the Monte Carlo estimation of the required probabilities, which essentially preserves the approximation quality. Recall that, due to the #P-completeness of two-terminal reliability, the Monte Carlo estimation of connection probabilities is computationally intensive for very small values of these probabilities. A key feature of our approximation algorithms is that they only require the estimation of probabilities not much smaller than the optimal value of the objective functions. In the case of the MCP problem, this is achieved by a simple adaptation of an existing approximation algorithm for the k-center problem [19], while for the ACP problem our strategy to avoid estimating small connection probabilities is novel.

To the best of our knowledge, ours are the first clustering algorithms for uncertain graphs that are fully parametric in the number of desired clusters while offering provable guarantees with respect to the optimal solution, together with efficient implementations. While the theoretical bounds on the approximation are somewhat loose, especially for small values of the optimum, we report the results of a number of experiments showing that in practice the quality of the clusterings returned by our algorithms is very high. In particular, we perform an experimental comparison of our algorithms with MCL which, as discussed above, is widely used in the context of uncertain graphs. We also compare with a naive adaptation of a classic k-center algorithm to verify that such adaptations lead to poor results, prompting for the development of specialized algorithms. We run our experiments on uncertain graphs derived from PPI networks and on a large collaboration graph derived from DBLP, finding that our algorithms identify good clusterings with respect to both our as well as other metrics, while a clustering strategy that is not specifically designed for uncertain graphs, as the one employed by MCL, may in some cases provide a very poor clustering. Moreover, on PPI networks, we evaluate the performance of our algorithms in predicting protein complexes, finding that we can obtain results comparable with state-of-the-art solutions.

The rest of the paper is organized as follows. In Section 2 we define basic concepts regarding uncertain graphs and formalize the MCP and ACP problems. Also, we prove a form of triangle inequality for the connection probability measure and discuss the NP-hardness of the two problems. In Section 3 we describe and analyze our clustering algorithms. In Section 4 we show how to integrate the Monte Carlo probability estimation within the algorithms while maintaining comparable approximation guarantees. Section 5 reports the results of the experiments. Finally, Section 6 offers some concluding remarks and discusses possible avenues of future research.

2. PRELIMINARIES

Let $\mathcal{G} = (V, E, p)$ be an uncertain graph, as defined in the introduction. In accordance with the established notation used in previous work, we write $G \sqsubseteq \mathcal{G}$ to denote that G is a possible world of $\mathcal{G}$. Given two nodes $u, v \in V$, the probability that they are connected (an event denoted as $u \sim v$) in a random possible world can be defined as

$$\Pr(u \sim v) = \sum_{G \sqsubseteq \mathcal{G}} \Pr(G)\mathbf{I}_G(u, v),$$

where

$$\mathbf{I}_G(u, v) = \begin{cases} 1 & \text{if } u \sim v \text{ in } G \\ 0 & \text{otherwise} \end{cases}$$

We refer to $\Pr(u \sim v)$ as the *connection probability* between u and v in $\mathcal{G}$. The uncertain graphs we consider in this paper, hence their possible worlds, are undirected and, except for the edge probabilities, no weights are attached to their nodes/edges.

Given an integer k, with $1 \leq k < n$, a *k-clustering* of $\mathcal{G}$ is a partition of V into k *clusters* $C_1, \ldots, C_k$ and a set of *centers* $c_1, \ldots, c_k$ with $c_i \in C_i$, for $1 \leq i \leq k$. We aim at clusterings where each node is well connected to its cluster center in a random possible world. To this purpose, for a k-clustering $\mathcal{C} = (C_1, \ldots, C_k; c_1, \ldots, c_k)$ of $\mathcal{G}$ we define the following two objective functions

$$\text{min-prob}(\mathcal{C}) = \min_{1 \leq i \leq k} \min_{v \in C_i} \Pr(c_i \sim v), \tag{1}$$

$$\text{avg-prob}(\mathcal{C}) = (1/n) \sum_{1 \leq i \leq k} \sum_{v \in C_i} \Pr(c_i \sim v). \tag{2}$$

Definition 1. Given an uncertain graph $\mathcal{G}$ with n nodes and an integer k, with $1 \leq k < n$, the *Minimum Connection Probability (MCP)* (resp., *Average Connection Probability (ACP)*) problem requires to determine a k-clustering $\mathcal{C}$ of $\mathcal{G}$ with maximum *min-prob*($\mathcal{C}$) (resp., *avg-prob*($\mathcal{C}$)).

By defining the distance between two nodes $u, v \in V$ as $d(u, v) = \ln(1/\Pr(u \sim v))$, with the understanding that $d(u, v) = \infty$ if $\Pr(u \sim v) = 0$, it is easy to see that the k-clustering that maximizes the objective function given in Equation (1) (resp., (2)) also minimizes the maximum distance (resp., the average distance) of a node from its cluster center. Therefore, the MCP and ACP problems can be reformulated as instances of the well-known NP-hard k-center and k-median problems [35], which makes the former the direct counterparts of the latter in the realm of uncertain graphs. However, objective functions that exercise alternative combinations of minimization and averaging of connection probabilities are in fact possible, and we leave their exploration as an interesting open problem.

While the k-center/median problems are NP-hard even when the distance function defines a metric space, thus, in particular, satisfying the triangle inequality (i.e, $d(u, z) \leq d(u, v) + d(v, z)$), this assumption is crucially exploited by most approximation strategies known in the literature. So, in order to port these strategies to the context of uncertain graphs, we need to show that the distances derived from the connection probabilities, as explained above, satisfy the triangle inequality. This is equivalent to showing that for any three nodes u, v, z, $\Pr(u \sim z) \geq \Pr(u \sim v) \cdot \Pr(v \sim z)$, which we prove below.

Fix an arbitrary edge $e \in E$ and let $A(e)$ be the event: "edge e is present". We need the following technical lemma.

LEMMA 1. *We have* $\Pr(x \sim y | A(e)) \geq \Pr(x \sim y | \neg A(e))$ *for any pair* $x, y \in V$.

PROOF. Let $\mathcal{G}_e^{x,y}$ (resp., $\mathcal{G}_{\neg e}^{x,y}$) be the set of possible worlds where $x \sim y$, and edge e is present (resp., not present). We have that

$$\Pr(x \sim y | A(e)) = \sum_{G \sqsubseteq \mathcal{G}_e^{x,y}} \Pr(G)/p(e),$$

$$\Pr(x \sim y | \neg A(e)) = \sum_{G \sqsubseteq \mathcal{G}_{\neg e}^{x,y}} \Pr(G)/(1 - p(e)).$$

The lemma follows by observing that for any graph G in $\mathcal{G}_{\neg e}^{x,y}$ the same graph with the addition of e belongs to $\mathcal{G}_e^{x,y}$, and the corresponding terms in the two summations are equal. □

THEOREM 1. *For any uncertain graph* $\mathcal{G} = (V, E, p)$ *and any triplet* $u, v, z \in V$, *we have:*

$$\Pr(u \sim z) \geq \Pr(u \sim v) \cdot \Pr(v \sim z).$$

PROOF. The proof proceeds by induction on the number k of *uncertain edges*, that is, edges $e \in E$ with $p(e) > 0$ and $p(e) < 1$. Fix three nodes $u, v, z \in V$. The base case $k = 0$ is trivial: in this case, the uncertain graph is deterministic and for each pair of nodes $x, y \in V$, $\Pr(x \sim y)$ is either 1 or 0, which implies that when $\Pr(u \sim v) \cdot \Pr(v \sim z) = 1$, then $\Pr(u \sim z) = 1$ as well. Suppose that the property holds for uncertain graphs with at most k uncertain edges, with $k \geq 0$, and consider an uncertain graph $\mathcal{G} = (V, E, p)$ with $k+1$ uncertain edges. Fix an arbitrary uncertain edge $e \in E$ and let $A(e)$ denote the event that edge e is present. For any two arbitrary nodes $x, y \in V$, we can write

$$\begin{aligned}\Pr(x \sim y) &= \Pr(x \sim y | A(e)) \cdot p(e) \\ &\quad + \Pr(x \sim y | \neg A(e)) \cdot (1 - p(e)) \\ &= (\Pr(x \sim y | A(e)) - \Pr(x \sim y | \neg A(e))) \cdot p(e) \\ &\quad + \Pr(x \sim y | \neg A(e)).\end{aligned}$$

By Lemma 1, the term multiplying $p(e)$ in the above expression is nonnegative. As a consequence, we have that

$$\Pr(u \sim v) \cdot \Pr(v \sim z) - \Pr(u \sim z) = A \cdot (p(e))^2 + B \cdot p(e) + C,$$

for some constants A, B, C independent of $p(e)$, with $A \geq 0$. Therefore, the maximum value of $\Pr(u \sim v) \cdot \Pr(v \sim z) - \Pr(u \sim z)$, as a function of $p(e)$, is attained for $p(e) = 0$ or $p(e) = 1$. Since in either case the number of uncertain edges is decremented by one, by the inductive hypothesis, the difference must yield a nonpositive value, hence the theorem follows. □

A fundamental primitive required for obtaining the desired clustering is the estimation of $\Pr(u \sim v)$ for any two nodes $u, v \in V$. While the exact computation of $\Pr(u \sim v)$ is $\#P$-complete [4], for reasonably large values of this probability a very accurate estimate can be obtained through Monte Carlo sampling. More precisely, for $r > 0$ let $G_1, \ldots, G_r$ be r sample possible worlds drawn independently at random from $\mathcal{G}$. For any pair of nodes u and v we can define the following estimator

$$\tilde{p}(u, v) = \frac{1}{r} \sum_{i=1}^{r} \mathbf{I}_{G_i}(u, v) \tag{3}$$

It is easy to see that $\tilde{p}(u, v)$ is an unbiased estimator of $\Pr(u \sim v)$. Moreover, by taking

$$r \geq \frac{3 \ln \frac{2}{\delta}}{\varepsilon^2 \Pr(u \sim v)} \tag{4}$$

samples, we have that $\tilde{p}(u, v)$ is an (ε, δ)-approximation of $\Pr(u \sim v)$, that is,

$$\Pr\left(\frac{|\tilde{p}(u, v) - \Pr(u \sim v)|}{\Pr(u \sim v)} \leq \varepsilon \right) \geq 1 - \delta \tag{5}$$

(e.g., see [27, Theorem 10.1]). This approach is very effective when $\Pr(u \sim v)$ is not very small. However, when $\Pr(u \sim v)$

is small (i.e., it approaches 0), the number of samples, hence the work, required to attain an accurate estimation becomes prohibitively large.

Even if the probabilities $\Pr(u \sim v)$ were provided by an oracle (i.e., they could be computed efficiently), the MCP problem remains computationally difficult. Indeed, consider the following decision problem: given an uncertain graph $\mathcal{G} = (V, E, p)$, an oracle for estimating pairwise connection probabilities, an integer $k \geq 1$, and $\hat{p}$ with $0 \leq \hat{p} \leq 1$, is there a k-clustering $\mathcal{C}$ such that min-prob$(\mathcal{C}) \geq \hat{p}$? We have:

THEOREM 2. *The above decision problem is NP-hard.*

The proof of Theorem 2, which is fairly technical, is based on a reduction from set cover and is omitted for brevity. (The proof can be found in [10].) We remark that the NP-hardness of our clustering problem on uncertain graphs does not follow immediately from the transformation of connection probabilities into distances mentioned earlier, which yields instances of the standard NP-hard k-center clustering problem, since such a transformation only shows that our problem is a restriction of the latter, where distances have the extra constraint to be derived from connection probabilities in the underlying uncertain graph.

We conjecture that a similar hardness result can be proved for the decision version of the ACP problem. Evidence in this direction is provided by the fact that by modifying both the MCP and ACP problems to feature a parametric upper limit on the lengths of the paths contributing to the connection probabilities, a variant which we study in Section 3.4, NP-hardness results for both the modified problems follow straightforwardly (i.e., when paths of length at most 1 are considered) from the hardness of k-center and k-median clustering.

3. CLUSTERING ALGORITHMS

A natural approach to finding good solutions for the MCP and ACP problems would be to resort to the well-known approximation strategies for the distance-based counterparts of these problems [31]. However, straightforward implementations of these strategies may require the computation of exact connection probabilities (to be transformed into distances) between arbitrary pairs of nodes, which can in principle be rather small. As an example, the popular k-center clustering strategy devised in [17] relies on the iterated selection of the next center as the *farthest* point from the set of currently selected ones, which corresponds to the determination of the node featuring the smallest connection probability to any node in the set, when adapted to the uncertain graph scenario. As we pointed out in the previous section, the exact computation of connection probabilities, especially if very small, is a computationally hard task. Therefore, for uncertain graphs we must resort to clustering strategies that are robust to approximations and try to avoid altogether the estimation of very small connection probabilities.

To address the above challenge, in Subsection 3.1 we introduce a useful primitive that, given a threshold q on the connection probability, returns a partial k-clustering of an uncertain graph $\mathcal{G}$ where the clusters cover a maximal subset of nodes, each connected to its cluster center with probability at least q, while all other nodes, deemed *outliers*, remain uncovered. In Subsections 3.2 and 3.3 we use such a primitive to derive approximation algorithms for the MCP and ACP problems, respectively, which feature provable guarantees on the quality of the approximation and lower bounds on the value of the connection probabilities that must ever be estimated. We also show how the approximation guarantees of the proposed algorithms change when connection probabilities are defined only with respect to paths of limited length.

All algorithms presented in this section take as input an uncertain graph $\mathcal{G} = (V, E, p)$ with n nodes, and assume the existence of an oracle that given two nodes $u, v \in V$ returns $\Pr(u \sim v)$. In Section 4 we will discuss how to integrate the Monte Carlo estimation of the connection probabilities within our algorithms.

3.1 Partial clustering

A *partial k-clustering* $\mathcal{C} = (C_1, \ldots, C_k; c_1, \ldots, c_k)$ of $\mathcal{G}$ is a partition of a subset of V into k clusters $C_1, \ldots, C_k$ (i.e., $\cup_{i=1,k} C_i \subseteq V$), where each cluster C_i is centered at $c_i \in C_i$, for $1 \leq i \leq k$. We can still define min-prob$(\mathcal{C})$ as in Equation 1 with the understanding that the uncovered nodes (i.e., $V - \cup_{i=1,k} C_i$) are not accounted for in min-prob$(\mathcal{C})$. In what follows, the term *full* k-clustering or, simply, k-clustering will refer only to a k-clustering covering all nodes.

The following algorithm, called MIN-PARTIAL (Algorithm 1 in the box), computes a partial k-clustering $\mathcal{C}$ of $\mathcal{G}$ with min-prob$(\mathcal{C}) \geq q$ covering a maximal set of nodes, in the sense that all nodes uncovered by the clusters have probability less than q of being connected to any of the cluster centers. The algorithm is based on a generalization of the strategy introduced in [11] and uses two design parameters, α and $\bar{q}$, where $\alpha \geq 1$ is an integer and $\bar{q} \in [q, 1]$, which are employed to exercise suitable tradeoffs between performance and approximation quality. In each of the k iterations, MIN-PARTIAL picks a suitable new center as follows. Let V' denote the nodes connected with probability less than q to the set of centers S selected so far. In the iteration, the algorithm selects an arbitrary set T of α nodes from V' (or all such nodes, if they are less than α) and picks as next center the node $v \in T$ that maximizes the number of nodes $u \in V'$ with $\Pr(u \sim v) \geq \bar{q}$. (The role of parameters α and $\bar{q}$ will be evident in the following subsections.) At the end of the k iterations, it returns the clustering defined by the best assignment of the covered nodes to the k selected centers. Namely, cluster C_i will consist of all covered nodes u such that $c_i = \arg\max_{c \in S}\{\Pr(u \sim c)\}$ (denoted as $c_i = c(u, S)$ in the pseudocode). It easily follows that for each $u \in C_i$, $\Pr(u \sim c_i) \geq q$.

3.2 MCP clustering

We now turn the attention to the MCP problem. The following lemma shows that if Algorithm MIN-PARTIAL is provided with a suitable guess q for the minimum connection probability, then the returned clustering covers all nodes and is a good solution to the MCP problem. Let $p_{\mathrm{opt-min}}(k)$ be the maximum value of min-prob$(\mathcal{C})$ over all full k-clusterings $\mathcal{C}$ of $\mathcal{G}$. (Observe that $p_{\mathrm{opt-min}}(k) > 0$ if and only if $\mathcal{G}$ has at most k connected components and, for convenience, we assume that this is always the case.)

LEMMA 2. *For any $q \leq p^2_{\mathrm{opt-min}}(k)$, $\alpha \geq 1$, and $\bar{q} \in [q, 1]$ we have that the k-clustering $\mathcal{C}$ returned by* MIN-PARTIAL$(\mathcal{G}, k, q, \alpha, \bar{q})$ *covers all nodes.*

PROOF. Consider an optimal k-clustering $\hat{\mathcal{C}} = (\hat{C}_1, \ldots, \hat{C}_k; \hat{c}_1, \ldots, \hat{c}_k)$ of $\mathcal{G}$, with $V = \cup_{i=1,k} \hat{C}_i$ and

Algorithm 1: MIN-PARTIAL$(\mathcal{G}, k, q, \alpha, \bar{q})$

$S \leftarrow \emptyset$; ▷ Set of centers
$V' \leftarrow V$;
for $i \leftarrow 1$ **to** k **do**
  select an arbitrary $T \subseteq V'$ with $|T| = \min\{\alpha, |V'|\}$;
  for $(v \in T)$ **do** $M_v \leftarrow \{u \in V' : \Pr(u \sim v) \geq \bar{q}\}$;
  $c_i \leftarrow \arg\max_{v \in T} |M_v|$;
  $S \leftarrow S \cup \{c_i\}$;
  $V' \leftarrow V' - \{u \in V' : \Pr(u \sim c_i) \geq q\}$;
end
if $(|S| < k)$ **then**
  add $k - |S|$ arbitrary nodes of $V - S$ to S;
$S \leftarrow \{c_1, \ldots, c_k\}$;
for $i \leftarrow 1$ **to** k **do** $C_i \leftarrow \{u \in V - V' : c(u, S) = c_i\}$;
return $\mathcal{C} = (C_1, \ldots, C_k; c_1, \ldots, c_k)$;

Algorithm 2: MCP$(\mathcal{G}, k, \gamma)$

$q \leftarrow 1$;
while *true* **do**
  $\mathcal{C} \leftarrow$ MIN-PARTIAL$(\mathcal{G}, k, q, 1, q)$;
  if $\mathcal{C}$ *covers all nodes* **then return** $\mathcal{C}$;
  else $q \leftarrow q/(1+\gamma)$;
end

min-prob$(\hat{\mathcal{C}}) = p_{\mathrm{opt-min}}(k)$. Let c_i be the center added to S in the i-th iteration of the **for** loop of MIN-PARTIAL, and let $\hat{C}_{j_i}$ be the cluster in $\hat{\mathcal{C}}$ which contains c_i, for every $i \geq 1$. By Theorem 1 we have that for every node $v \in \hat{C}_{j_i}$

$$\Pr(c_i \sim v) \geq \Pr(c_i \sim \hat{c}_{j_i}) \cdot \Pr(\hat{c}_{j_i} \sim v) \geq p^2_{\mathrm{opt-min}}(k) \geq q$$

Therefore, at the end of the i-th iteration of the **for** loop, V' cannot contain nodes of $\hat{C}_{j_i}$. An easy induction shows that at the end of the **for** loop, V' is empty. □

Based on the result of the lemma, we can solve the MCP problem by repeatedly running MIN-PARTIAL with progressively smaller guesses of q, starting from $q = 1$ and decreasing q by a factor $(1+\gamma)$, for a suitable parameter $\gamma > 0$, at each run, until a clustering covering all nodes is obtained. We refer to this algorithm as MCP (Algorithm 2 in the box). The following theorem is an immediate consequence of Lemma 2.

THEOREM 3. *Algorithm 2 requires at most* $\lfloor 2\log_{1+\gamma}(1/p_{\mathrm{opt-min}}(k)) \rfloor + 1$ *executions of* MIN-PARTIAL, *and returns a* k*-clustering* $\mathcal{C}$ *with*

$$\textit{min-prob}(\mathcal{C}) \geq \frac{p^2_{\mathrm{opt-min}}(k)}{(1+\gamma)}.$$

It is easy to see that all connection probabilities $\Pr(u \sim v)$ used in Algorithm 2 are not smaller than $p^2_{\mathrm{opt-min}}(k)/(1+\gamma)$. Also, we observe that once q becomes sufficiently small to ensure the existence of a full k-clustering, a binary search between the last two guesses for q can be performed to get a higher minimum connection probability.

3.3 ACP clustering

In order to compute good solutions to the ACP problem we resort again to the computation of partial clusterings. For a given connection probability threshold $q \in (0, 1]$, define t_q as the minimum number of nodes left uncovered by any partial k-clustering $\mathcal{C}$ of $\mathcal{G}$ with min-prob$(\mathcal{C}) \geq q$. It is easy to argue that t_q is a non-decreasing function of q. Observe that any partial k-clustering can be "completed", i.e., turned into a full k-clustering, by assigning the uncovered nodes arbitrarily to the available clusters (possibly with connection probabilities to the cluster centers equal to 0), and that $q(n - t_q)/n$ is a lower bound to the average connection probability of such a full k-clustering. Let $p_{\mathrm{opt-avg}}(k)$ be the maximum value of avg-prob$(\mathcal{C})$ over all k-clusterings $\mathcal{C}$ of $\mathcal{G}$. The following lemma shows that for a suitable q, the value $q(n - t_q)/n$ is not much smaller than $p_{\mathrm{opt-avg}}(k)$.

LEMMA 3. *There exists a value* $q \in (0, 1]$ *such that*

$$q \cdot \frac{n - t_q}{n} \geq \frac{p_{\mathrm{opt-avg}}(k)}{H(n)},$$

where $H(n) = \sum_{i=1}^{n}(1/i) = \ln n + O(1)$ *is the* n*-th harmonic number.*

PROOF. Let $\hat{\mathcal{C}}$ be the k-clustering of $\mathcal{G}$ which maximizes the average connection probability. Let $p_0 \leq p_1 \leq \cdots \leq p_{n-1}$ the connection probabilities of the n nodes to their cluster centers in $\hat{\mathcal{C}}$, sorted by non-decreasing order, and note that avg-prob$(\hat{\mathcal{C}}) = (1/n)\sum_{i=0}^{n-1} p_i$. It is easy to argue that for each $0 \leq i < n$ there exists a partial k-clustering of $\mathcal{G}$ which covers $n - i$ nodes and where each covered node is connected to its cluster center with probability at least p_i. This implies that $t_{p_i} \leq i$. We claim that

$$\max_{i=0,\ldots,n-1} p_i \frac{n-i}{n} \geq \frac{p_{\mathrm{opt-avg}}(k)}{H(n)}.$$

If this were not the case we would have

$$\begin{aligned} p_{\mathrm{opt-avg}}(k) &= \frac{1}{n}\sum_{i=0}^{n-1} p_i \\ &< \frac{p_{\mathrm{opt-avg}}(k)}{H(n)} \sum_{i=0}^{n-1} \frac{1}{n-i} = p_{\mathrm{opt-avg}}(k), \end{aligned}$$

which is impossible. Therefore, there must exist an index $i \in [0, n-1]$ such that

$$p_i \frac{n - t_{p_i}}{n} \geq p_i \frac{n-i}{n} \geq \frac{p_{\mathrm{opt-avg}}(k)}{H(n)}. \quad \square$$

Based on the above lemma, we can obtain an approximate solution for the ACP problem by seeking partial k-clusterings which strike good tradeoffs between the minimum connection probability and the number of uncovered points. The next lemma shows that Algorithm MIN-PARTIAL can indeed provide these partial clusterings.

LEMMA 4. *For any* $q \in (0, 1]$, *we have that the partial* k*-clustering* $\mathcal{C}$ *returned by* MIN-PARTIAL$(\mathcal{G}, k, q^3, n, q)$ *covers all but* t_q *nodes.*

PROOF. The proof can be obtained by rephrasing the proof of the 3-approximation result for the *robust k-center problem* in [11, Theorem 3.1] in terms of connection probabilities rather than distances. □

We are now ready to describe our approximation algorithm for the ACP problem, which we refer to as ACP. The idea is to run MIN-PARTIAL so to obtain partial k-clusterings with t_q

Algorithm 3: ACP$(\mathcal{G}, k, \gamma)$

$\mathcal{C} \leftarrow$ MIN-PARTIAL$(\mathcal{G}, k, 1, n, 1)$;
$\phi_{\text{best}} \leftarrow (1/n)\sum_{u \in V} p_{\mathcal{C}}(u)$;
$\mathcal{C}_{\text{best}} \leftarrow$ any full k-clustering completing $\mathcal{C}$;
$q \leftarrow 1/(1+\gamma)$;
while $(q^3 \geq \phi_{\text{best}})$ **do**
  $\mathcal{C} \leftarrow$ MIN-PARTIAL$(\mathcal{G}, k, q^3, n, q)$;
  $\phi \leftarrow (1/n)\sum_{u \in V} p_{\mathcal{C}}(u)$;
  if $(\phi \geq \phi_{\text{best}})$ **then**
    $\phi_{\text{best}} \leftarrow \phi$;
    $\mathcal{C}_{\text{best}} \leftarrow$ any full k-clustering completing $\mathcal{C}$;
  end
  else $q \leftarrow q/(1+\gamma)$;
end
return $\mathcal{C}_{\text{best}}$

uncovered nodes, for progressively smaller values of q. For each such value of q, MIN-PARTIAL is employed to compute a partial k-clustering $\mathcal{C}$ where at most t_q nodes remain uncovered and where each covered node is connected to its cluster center with probability at least q^3. If $\mathcal{C}$ has the potential to provide a full k-clustering with higher average connection probability than those previously found, it is turned into a full k-clustering by assigning the uncovered nodes to arbitrary clusters. The algorithm halts when further smaller guesses of q cannot lead to better clusterings. The pseudocode (Algorithm 3 in the box) uses the following notation. For a (partial) k-clustering $\mathcal{C}$ and a node $u \in V$, $p_{\mathcal{C}}(u)$ denotes the connection probability of u to the center of its assigned cluster, if any, and set $p_{\mathcal{C}}(u) = 0$ otherwise. We have:

THEOREM 4. *Algorithm 3 returns a k-clustering $\mathcal{C}$ with*

$$\textit{avg-prob}(\mathcal{C}) \geq \left(\frac{p_{\text{opt}-\text{avg}}(k)}{(1+\gamma)H(n)}\right)^3,$$

where $H(n)$ is the n-th harmonic number, and requires at most $\left\lfloor \log_{1+\gamma}(H(n)/p_{\text{opt}-\text{avg}}(k)) \right\rfloor + 1$ executions of MIN-PARTIAL.

PROOF. Note that the while loop maintains, as an invariant, the relation avg-prob$(\mathcal{C}_{\text{best}}) \geq \phi_{\text{best}}$. Hence, this relation holds at the end of the algorithm when the k-clustering $\mathcal{C}_{\text{best}}$ is returned. Let $q^* \in (0,1]$ be a value such that

$$q^* \cdot \frac{n - t_{q^*}}{n} \geq \frac{p_{\text{opt}-\text{avg}}(k)}{H(n)}.$$

The existence of q^* is ensured by Lemma 3. If the while loop ends when $q > q^*$, then

$$\begin{aligned}\phi_{\text{best}} &> q^3 > (q^*)^3 \geq \left(\frac{n}{n-t_{q^*}}\frac{p_{\text{opt}-\text{avg}}(k)}{H(n)}\right)^3 \\ &\geq \left(\frac{p_{\text{opt}-\text{avg}}(k)}{H(n)}\right)^3.\end{aligned}$$

If instead q becomes $\leq q^*$, consider the first iteration of the while loop when this happens, that is when $q^*/(1+\gamma) < q \leq q^*$ and let $\mathcal{C}$ be the partial k-clustering computed in the iteration. By Lemma 4, at most t_q nodes are not covered by $\mathcal{C}$ and since t_q is non-decreasing, as observed before, we have that $t_q < t_{q^*}$. This implies that the value ϕ derived from $\mathcal{C}$ (hence, ϕ_{best} at the end of the iteration) is such that

$$\begin{aligned}\phi &> q^3 \cdot \frac{n-t_q}{n} \geq \left(\frac{q^*}{1+\gamma}\right)^3 \cdot \frac{n-t_{q^*}}{n} \\ &\geq \left(\frac{n}{n-t_{q^*}}\frac{p_{\text{opt}-\text{avg}}(k)}{(1+\gamma)H(n)}\right)^3 \cdot \frac{n-t_{q^*}}{n} \\ &\geq \left(\frac{p_{\text{opt}-\text{avg}}(k)}{(1+\gamma)H(n)}\right)^3.\end{aligned}$$

In all cases, the average connection probability of the returned clustering satisfies the stated bound. As for the upper bound on the number of iterations of the while loop, we proved above that as soon as q falls in the interval $(q^*/(1+\gamma), q^*]$ we have $\phi_{\text{best}} \geq (p_{\text{opt}-\text{avg}}(k)/((1+\gamma)H(n)))^3$, hence, from that point on, q cannot become smaller than $p_{\text{opt}-\text{avg}}(k)/((1+\gamma)H(n)) < q^*$. This implies that $\left\lfloor \log_{1+\gamma}(H(n)/p_{\text{opt}-\text{avg}}(k)) \right\rfloor + 1$ iterations of the while loop are executed overall. □

It is easy to see that all connection probabilities $\Pr(u \sim v)$ that Algorithm 3 needs to compute in order to be correct are not smaller than $(p_{\text{opt}-\text{avg}}(k)/((1+\gamma)H(n)))^3$.

We remark that while the theoretical approximation ratios attained by our algorithms for both the MCP and ACP problems appear somewhat weak, especially for small values of $p_{\text{opt}-\text{min}}$ and $p_{\text{opt}-\text{avg}}$, we will provide experimental evidence (see Section 5) that, in practical scenarios where connection probabilities are not too small, they return good-quality clusterings and, by avoiding the estimation of small connection probabilities, they run relatively fast.

3.4 Limiting the path length

The algorithms described in the preceding subsections can be run by setting a limit on the length of the paths that contribute to the connection probability between two nodes. As mentioned in the introduction, this feature may be useful in application scenarios where the similarity between two nodes diminishes sharply with their topological distance regardless of their connection probability.

For a fixed integer d, with $1 \leq d < n$, we define $\Pr(u \overset{d}{\sim} v) = \sum_{G \sqsubseteq \mathcal{G}} \Pr(G)\mathbf{I}_G(u,v;d)$, where $\mathbf{I}_G(u,v;d)$ is 1 if u is at distance at most d from v in G, and 0 otherwise. In the following, we refer to $\Pr(u \overset{d}{\sim} v)$ as the *d-connection probability* between u and v. By easily adapting the proofs of Lemma 1 and Theorem 1, it can be shown that for any pair of distances d_1, d_2, with $d \geq d_1 + d_2$, and for any triplet $u, v, z \in V$, it holds that

$$\Pr(u \overset{d}{\sim} z) \geq \Pr(u \overset{d_1}{\sim} v) \cdot \Pr(v \overset{d_2}{\sim} z).$$

We now reconsider the MCP and ACP problems under paths of limited depth. For a k-clustering $C_1, \ldots, C_k$ with centers $\{c_1, \ldots, c_k\}$, we define the objective functions

$$\text{min-prob}_d(\mathcal{C}) = \min_{1 \leq i \leq k} \min_{v \in C_i} \Pr(c_i \overset{d}{\sim} v), \tag{6}$$

$$\text{avg-prob}_d(\mathcal{C}) = (1/n) \sum_{1 \leq i \leq k} \sum_{v \in C_i} \Pr(c_i \overset{d}{\sim} v), \tag{7}$$

and let $p_{\text{opt}-\text{min}}(k,d)$ and $p_{\text{opt}-\text{avg}}(k,d)$, respectively, be the maximum values of these two objective functions over all k-clusterings. Algorithms 1, 2 and 3 can all be rephrased

by imposing limited path lengths in the estimation of connection probabilities so to obtain the results stated in the following two theorems, whose proofs are omitted for lack of space and can be found in [10].

THEOREM 5. *Suppose that* $p_{opt\text{-}min}(k, \lfloor d/2 \rfloor) > 0$. *When run with d-connection probabilities, Algorithm 2 requires at most* $\lfloor 2\log_{1+\gamma}(1/p_{\text{opt}-\min}(k, \lfloor d/2 \rfloor)) \rfloor + 1$ *executions of* MIN-PARTIAL*, and returns a k-clustering* $\mathcal{C}$ *with*

$$\textit{min-prob}_d(\mathcal{C}) \geq \frac{p^2_{\text{opt}-\min}(k, \lfloor d/2 \rfloor)}{(1+\gamma)}.$$

THEOREM 6. *When run with d-connection probabilities, Algorithm 3 returns a k-clustering* $\mathcal{C}$ *with*

$$\textit{avg-prob}_d(\mathcal{C}) \geq \left(\frac{p_{\text{opt}-\text{avg}}(k, \lfloor d/3 \rfloor)}{(1+\gamma)H(n)} \right)^3,$$

where $H(n)$ *is the n-th harmonic number, and requires at most* $\left\lfloor \log_{1+\gamma}(H(n)/p_{\text{opt}-\text{avg}}(k, \lfloor d/3 \rfloor)) \right\rfloor + 1$ *executions of* MIN-PARTIAL.

We remark that the assumption $p_{\text{opt-min}}(k, \lfloor d/2 \rfloor) > 0$ in Theorem 5 is required to ensure that, in Algorithm 2, a suitable guess for q is reached in a finite number of iterations. Termination is instead always guaranteed for Algorithm 3 since $p_{\text{opt}-\text{avg}}(k, \lfloor d/3 \rfloor) \geq k/n > 0$, for every $d \geq 0$.

4. IMPLEMENTING THE ORACLE

In the previous section we assumed that the probabilities $\Pr(u \sim v)$ could be obtained exactly from an oracle. In practice, the estimation of these probabilities is the most critical part for the efficient implementation of our algorithms. In this section, we show how to integrate the Monte Carlo sampling method for the estimation of the connection probabilities within the algorithms described in Section 3, maintaining similar guarantees on the quality of the returned clusterings. The basic idea of our approach is to adjust the number of samples dynamically during the execution of the algorithms, based on safe guesses of the probabilities that need to be estimated.

For ease of presentation, throughout this section we assume that lower bounds to $p^2_{\text{opt}-\min}(k)$ and to $(p_{\text{opt}-\text{avg}}(k)/H(n))^3$ are available. We will denote both lower bounds by p_L, since it will be clear from the context which one is used. For example, these lower bounds can be obtained by observing that $p_{\text{opt}-\min}(k)$ is greater than or equal to the probability of the most unlikely world, and $p_{\text{opt}-\text{avg}}(k) \geq k/n$. In practice, p_L can be employed as a threshold set by the user to exclude, a priori, clusterings with low values of the objective function. In this case, if the algorithm does not find a clustering whose objective function is above the threshold, it terminates by reporting that no clustering could be found. Recall that $\tilde{p}(u, v)$ denotes the estimate of the probability $\Pr(u \sim v)$ obtained by sampling possible worlds (see Equation 3). Moreover, for a node $u \in V$ and a set of nodes $S \subset V$, we define $\tilde{c}(u, S) = \arg\max_{c \in S}\{\tilde{p}(c, u)\}$ as the function returning the node of S connected to u with the highest *estimated* probability. Similarly, we define $\tilde{\pi}(u, S) = \Pr(\tilde{c}(u, S) \sim u) = \max_{c \in S}\{\tilde{p}(c, u)\}$. We use $\varepsilon > 0$ to denote an approximation parameter to be fixed by the user.

4.1 Partial clustering

A key component of Algorithms MCP and ACP is the MIN-PARTIAL subroutine (Algorithm 1), whose input includes two thresholds q and $\bar{q}$ for the connection probabilities. Note that in each invocation of MIN-PARTIAL within MCP and ACP, we have that $\bar{q} \geq q$, and that only connection probabilities not smaller than q are needed. We implement MIN-PARTIAL as follows. Suppose that we estimate connection probabilities using a number r of samples, based on Equation (4), which ensures that any $\Pr(u \sim v) \geq q$ is estimated with relative error at most $\varepsilon/2$ with probability at least $1 - \delta$, where $\varepsilon, \delta \in (0, 1)$ are suitable values. Then, in each of the k iterations of the main for-loop of MIN-PARTIAL, a new center c is selected which maximizes the number of uncovered nodes u with $\tilde{p}(c, u) \geq (1 - \frac{\varepsilon}{2})\bar{q}$, and all nodes u with $\tilde{p}(c, u) \geq (1 - \frac{\varepsilon}{2})q$ are removed from the set V' of uncovered nodes. The following two subsections analyze the quality of the clusterings returned by Algorithms MCP and ACP when using this implementation of MIN-PARTIAL.

4.2 Implementation of MCP

Recall that Algorithm MCP invokes MIN-PARTIAL with a probability threshold q which is lowered at each iteration of its main while loop. Using the implementation of MIN-PARTIAL described before, this iterative adjustment of q corresponds to a progressive sampling strategy. In particular, if for each iteration of the while loop we use a number of samples

$$r = \left\lceil \frac{12}{q\varepsilon^2} \ln\left(2n^3\left(1 + \left\lfloor \log_{1+\gamma}\frac{1}{p_L} \right\rfloor\right)\right) \right\rceil, \qquad (8)$$

we obtain the following result.

THEOREM 7. *The implementation of* MCP *terminates after at most* $\lfloor 2\log_{1+\gamma}(1/p_{\text{opt}-\min}(k)) \rfloor + 1$ *iterations of the while loop and returns a clustering* $\tilde{\mathcal{C}}$ *with*

$$\textit{min-prob}(\tilde{\mathcal{C}}) \geq \frac{(1-\varepsilon)}{(1+\gamma)} p^2_{\text{opt}-\min}(k)$$

if $p^2_{\text{opt}-\min}(k) > p_L$*, with high probability.*

PROOF. Consider an arbitrary iteration of the while loop of MCP, and a pair of nodes $u, v \in V$. For

$$\delta = \frac{1}{n^3\left(1 + \left\lfloor \log_{1+\gamma}\frac{1}{p_L} \right\rfloor\right)},$$

we have that using the number of samples specified by Equation (8), the following properties for the estimate $\tilde{p}(u, v)$ hold:

- if $\Pr(u \sim v) > q$, then $\tilde{p}(u, v) < (1 - \frac{\varepsilon}{2})q$ with probability $< \delta$.
- if $\Pr(u \sim v) < (1 - \varepsilon)q$, then $\tilde{p}(u, v) \geq (1 - \frac{\varepsilon}{2})q$ with probability $< \delta$.

Moreover, note that MCP performs at most $1 + \lfloor \log_{1+\gamma}\frac{1}{p_L} \rfloor$ iterations of the while loop. Therefore, by union bound on the number of node pairs and the number of iterations, we have that the following holds with probability at least $1 - 1/n$: in each iteration of the while loop every node connected to some center with probability $\geq q$ is added to a cluster, and no cluster contains nodes whose connection probability to the center is $< (1 - \varepsilon)q$. Consider now the ℓ-th iteration,

with $\ell = \lfloor 2\log_{1+\gamma}(1/p_{\text{opt-min}}(k))\rfloor + 1$, in which we have $q \leq p^2_{\text{opt-min}}(k)$. In this iteration, the algorithm completes and returns a clustering. Since at the beginning of this iteration we have $q > p^2_{\text{opt-min}}(k)/(1+\gamma)$, by the above discussion we have that

$$\text{min-prob}(\tilde{\mathcal{C}}) \geq (1-\varepsilon)q \geq \frac{(1-\varepsilon)}{(1+\gamma)} p^2_{\text{opt-min}}(k)$$

with probability at least $1-1/n$, and the theorem follows □

We observe that, for constant γ, the overall number of samples required by our implementation is $O\left((1/(p_{\text{opt-min}}(k)\varepsilon)^2)(\log n + \log\log(1/p_L))\right)$. As we mentioned before, p_L can be safely set equal to the probability of the most unlikely world. In case this lower bound were too small, a progressive sampling schedule similar to the one adopted in [29] could be used where p_L is not required, which ensures that $O\left((1/(p_{\text{opt-min}}(k)\varepsilon)^2)(\log n + \log(1/p_{\text{opt-min}}(k)))\right)$ samples suffice.

4.3 Implementation of ACP

Algorithm ACP also uses a probability threshold q which is lowered at each iteration of its main while loop, and in each iteration it needs to estimate reliably probabilities that are at least q^3. Again, we use the implementation of MIN-PARTIAL described before and in each iteration of the while loop we set the number of samples as

$$r = \left\lceil \frac{12}{q^3\varepsilon^2} \ln\left(2n^3\left(1 + \left\lfloor \log_{1+\gamma}\frac{H(n)}{p_L}\right\rfloor\right)\right)\right\rceil. \qquad (9)$$

We have the following result, whose proof, analogous to that of Theorem 7, is omitted for brevity.

THEOREM 8. *The implementation of* ACP *terminates after at most* $\lfloor \log_{1+\gamma}(H(n)/p_{\text{opt-avg}}(k))\rfloor + 1$ *iterations of the while loop and returns a clustering* $\tilde{\mathcal{C}}$ *with*

$$\textit{avg-prob}(\tilde{\mathcal{C}}) \geq (1-\varepsilon)\left(\frac{p_{\text{opt-avg}}(k)}{(1+\gamma)H(n)}\right)^3$$

if $(p_{\text{opt-avg}}(k)/H(n))^3 \geq p_L$*, with high probability.*

We observe that, for constant γ, the overall number of samples required by our implementation is $O\left((1/(p^3_{\text{opt-avg}}(k)\varepsilon^2)(\log n + \log\log((\log n)/p_L))\right)$. Considering that we can safely set $p_L = k/n$, as mentioned at the beginning if the section, we conclude that $O\left((1/(p^3_{\text{opt-avg}}(k)\varepsilon^2)\log n\right)$ samples suffice.

5. EXPERIMENTS

We experiment with our clustering algorithms MCP and ACP along two different lines. First, in Subsection 5.1, we compare the quality of the obtained clusterings against those returned by well-established clustering approaches in the literature on four uncertain graphs derived by three PPI networks and a collaboration network. Then, in Subsection 5.2, we provide an example of applicability of uncertain clustering as a predictive tool to spot so-called *protein complexes* in one of the aforementioned PPI networks.

The characteristics of the four different graphs used in our experiments are summarized in Table 1. Three graphs are PPI networks, with different distributions of edge probabilities: `Collins` [12], mostly comprising high-probability edges; `Gavin` [15], where most edges are associated to low probabilities, and the CORE network introduced in [24] (`Krogan` in the following), which has one fourth of the edges with probability greater than 0.9, and the others almost uniformly distributed between 0.27 and 0.9. To exercise a larger spectrum of cluster granularities, we target clusterings only for the largest connected component of each graph. As a computationally more challenging instance, we also experiment with a large connected subgraph of the DBLP collaboration network with edge probabilities obtained with the same procedure of [30]: each node is an author, and two authors are connected by an edge if they are co-authors of at least one journal publication. The probability of such an edge is $1-\exp\{-x/2\}$, where x is the number of co-authored journal papers. Consequently, a single collaboration corresponds to an edge with probability 0.39, and 2 and 5 collaborations correspond to edges with probability 0.63 and 0.91, respectively. Roughly 80% of the edges have probability 0.39, 12% have probability 0.63 and the remaining 8% have a higher probability. While finding an accurate probabilistic model of the interactions between authors is beyond the scope of this paper, the intuition behind the choice of this distribution is that authors that are likely to collaborate again in the future share an edge with large probability.

Table 1: Graphs considered in our experiments. The number of nodes and the number of edges in the largest connected component are shown.

graph	nodes	edges
`Collins`	1004	8323
`Gavin`	1727	7534
`Krogan`	2559	7031
`DBLP`	636751	2366461

We implemented our algorithms in C++, with the Monte Carlo sampling of possible worlds performed in parallel using OpenMP. The code and data, along with instructions to reproduce the results presented in this section, are publicly available[1]. When running both MCP and ACP, we set $\gamma = 0.1$. To optimize the execution time, we set the probability threshold q of Algorithms 2 and 3 to $q_i = \max\{1-\gamma\cdot 2^i, p_L\}$ in iteration i, with $p_L = 10^{-4}$. Once q_i equals p_L or is such that the associated clustering covers all nodes, we perform a binary search between q_i and q_{i-1} to find the final probability guess, stopping when the ratio between the lower and upper bound is greater than $1-\gamma$. This procedure is essentially equivalent, up to constant factors in the final value of q, to decreasing q geometrically as is done in Algorithms 2 and 3, thus the guarantees of Theorems 7 and 8 hold. In the implementation of ACP, we decided to invoke MIN-PARTIAL with parameters $(\mathcal{G}, k, q, 1, q)$ rather than $(\mathcal{G}, k, q^3, n, q)$. While this setting does not guarantee the theoretical bounds stated in Theorem 8, the values were chosen after testing different combinations of the two parameters and finding that the chosen setting provides better time performance while still returning good quality clusterings in all tested scenarios. In particular, we found out that higher values of parameter α yielded similar scores, albeit with a lower variance. For both implementations, we verified that setting parameter γ, which essentially controls a

[1] `https://github.com/Cecca/ugraph`

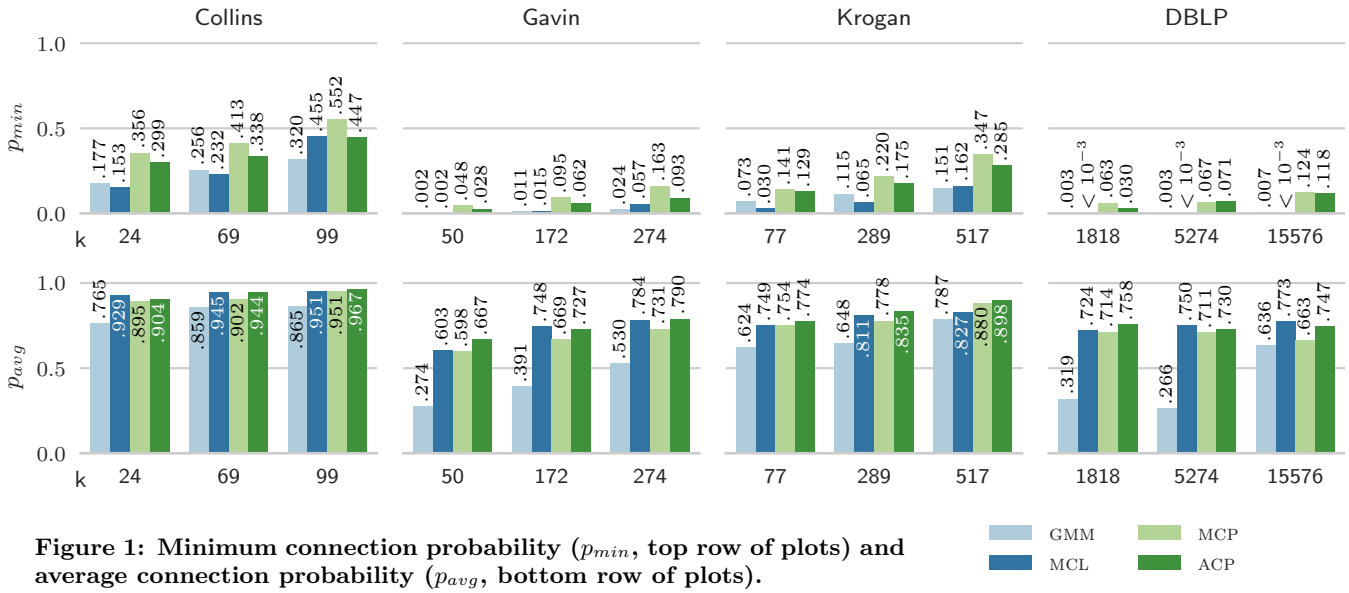

Figure 1: Minimum connection probability (p_{min}, top row of plots) and average connection probability (p_{avg}, bottom row of plots).

time/quality tradeoff, to values smaller than 0.1 increases the running time without increasing the quality of the returned clusterings significantly. Considering that the sample sizes defined in Section 4 are derived to tolerate very conservative union bounds, we verified that in practice starting the progressive sampling schedule from 50 samples always yields very accurate probability estimates. We omit the results of these preliminary experiments for lack of space. Both our code and MCL were compiled with GCC 5.4.0, and run on a Linux 4.4.0 machine equipped with an Intel I7 4-core processor and 18GB of RAM. Each figure we report was obtained as the average over at least 100 runs, with the exception of the bigger DBLP dataset, where only 5 runs were executed for practicality.

5.1 Comparison with other algorithms

A few remarks on the algorithms chosen for (or excluded from) the comparison with MCP and ACP presented in this subsection are in order. We do not compare with the clustering algorithm for uncertain graphs devised in [23] because it does not allow to control the number k of returned clusters, which is central in our setting. (A comparison between MCP and the algorithm by [23] is offered in the next subsection with respect to a specific predictive task.) Also, we were forced to exclude from the comparison the clustering algorithms of [25], explicitly devised for uncertain graphs, since the source code was not made available by the authors and the algorithms do not lend themselves to a straightforward implementation. Among the many clustering algorithms for deterministic weighted graphs available in the literature, we selected MCL [34] since it has been previously employed in the realm of uncertain graphs by using edge probabilities as weights. Finally, we compare with an adaptation of the k-center approximation strategy of [17], dubbed GMM, where the clustering of the uncertain graph is computed in k iterations by repeatedly picking the farthest node from the set of previously selected centers, using the shortest-path distances generated by setting the weight of any edge e to $w(e) = \ln(1/p(e))$. Considering the modest performance of GMM observed in our experiments, and the fact that it has been repeatedly stated in previous works [25, 30] that the application of shortest-path based deterministic strategies to the probabilistic context yields unsatisfactory results, we did not deem necessary to extend the comparison to other such strategies.

We base our comparison both on our defined cluster quality metrics and on other ones. Namely, for each clustering returned by the various algorithms, we compute the minimum connection probability of any node to its center[2] (denoted simply as p_{min}), and the average connection probability of all nodes to their respective centers (denoted as p_{avg}). Furthermore, we consider the inner Average Vertex Pairwise Reliability (also defined in [25]) which is the average connection probability of all pairs of nodes that are in the same cluster, namely,

$$\text{inner-AVPR} = \frac{\sum_{i=1}^{\tau} \sum_{u,v \in C_i} \Pr(u \sim v)}{\sum_{i=1}^{\tau} \binom{|C_i|}{2}},$$

and the outer Average Vertex Pairwise Reliability, which is the average connection probability of pairs of nodes in different clusters, namely,

$$\text{outer-AVPR} = \frac{\sum_{i=1}^{\tau} \sum_{u \in C_i, v \notin C_i} \Pr(u \sim v)}{\sum_{i=1}^{\tau} |C_i| \cdot |V \setminus C_i|}.$$

Intuitively, a good clustering in terms of connection probabilities exhibits a low outer-AVPR and a significantly higher inner-AVPR, indicating that each cluster completely encapsulates a region of high reliability. Inner-AVPR and outer-AVPR are akin to the internal and external *cluster density* measures used in the setting of deterministic graphs [31]. We also compare the algorithms in terms of their running time.

Recall from Section 1 that the number of clusters computed by MCL cannot be controlled accurately, but it is influenced by the inflation parameter. Therefore, for each graph in Table 1, we run MCL with inflation set to 1.2, 1.5, and 2.0 for protein networks, and 1.15, 1.2, and 1.3 for DBLP, so

[2]For MCL, when computing this metric we consider as cluster centers the *attractor nodes* as defined in [34].

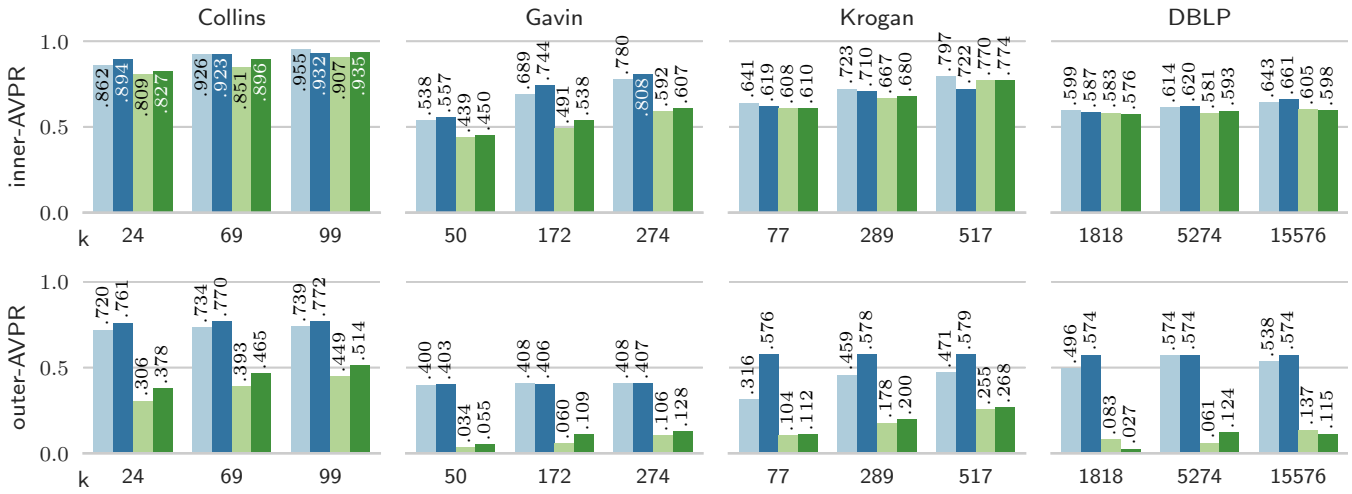

Figure 2: Inner and outer Average Vertex Pairwise Reliability. For the inner-AVPR metric higher is better, for the outer-AVPR metric lower is better.

to obtain a reasonable number of clusters. We then run the other algorithms with a target number k of clusters matching the granularity of the clustering returned by each MCL run. Note that, in terms of running time, this setup favors MCL: if we were instead given a target number of clusters, we would need to perform a binary search over the possible inflation values to make MCL match the target, and the running time for MCL would become the sum of the times over all these search trials.

As expected, with respect to the p_{min} metric (Figure 1, top) MCP is always better than all other algorithms. In particular, on the DBLP graph, both GMM and MCL find clusterings with p_{min} very close to zero ($< 10^{-3}$), meaning that there is at least one pair of nodes in the same cluster with almost zero connection probability. In contrast, MCP finds clusterings with p_{min} very close to 0.1 or larger. The inferior performance of GMM, a clustering algorithm aiming at optimizing an extremal statistic like p_{min} in a deterministic graph, provides evidence that naive adaptations of deterministic clustering algorithms to the probabilistic scenario struggle to find good solutions. Further evidence in this direction is provided by the fact that ACP yields higher values of p_{min} than both GMM and MCL.

For what concerns the p_{avg} metric (Figure 1, bottom), observe that, being an average, the metric hides the presence of low probability connections in the same cluster, which explains the much higher values returned by all algorithms compared to p_{min}. Somewhat surprisingly, we have that MCL and ACP have comparable performance. Recall, however, that MCL does not guarantee total control on the number of clusters, which makes ACP a more flexible tool. In all cases, the GMM algorithm finds clusters with a p_{avg} that is lower than the one obtained by the other algorithms. We observe that the experiments provide evidence that the actual values of p_{avg} obtained with ACP are arguably much higher than the theoretical bounds proved in Theorem 8, and we conjecture that this also holds for p_{min}. Consider now the inner- and outer-AVPR metrics (Figure 2, resp. top and bottom). For all graphs, the clusterings computed by MCP and ACP feature an inner-AVPR comparable to GMM and MCL, but a considerably lower outer-AVPR, which is a desirable property of a clustering, as mentioned earlier. Conversely, for a given graph and value of k, MCL and GMM compute clusterings where the inner- and outer-AVPR scores are similar. This fact suggests that the other clustering strategies are driven more by the topology of the graphs rather than by the connection probabilities.

As for the running times (Figure 3), GMM is almost always the fastest algorithm, due to its obliviousness to the possible world semantics that requires Monte Carlo sampling, and its running time grows linearly in k. On the other hand, MCL exhibits an opposite dependence on k since clusterings for low values of k, which are arguably more interesting in practical scenarios, are the most expensive to compute. Furthermore, we observed that the memory required by MCL increases sharply for small values of k (that is for small values of its inflation parameter): on the DBLP graph for $k < 1818$, MCL crashed after exhausting the memory available on the system, whereas our algorithm is very efficient for such values. Thanks to the use of progressive sampling, our algorithms feature a running time which is significantly better or comparable to MCL, depending on the granularity of the clustering. With respect to GMM our algorithms are slower, but they provide far better clusterings, as discussed above.

5.2 Clustering as a predictive tool

In PPI networks, proteins can be grouped in so-called *complexes*, that is, groups of proteins that stably interact with one another to perform various functions in a cell. Given a PPI network, a crucial problem is to predict protein pairs belonging to the same complex. Our specific benchmark is the Krogan graph, for which the authors published a clustering with 547 clusters obtained using MCL with a configuration of parameters that maximizes biological significance [24, Suppl. Table 10]. We consider a ground truth derived from the publicly available, hand-curated MIPS database of protein complexes [26, 20] as used in [24]. For

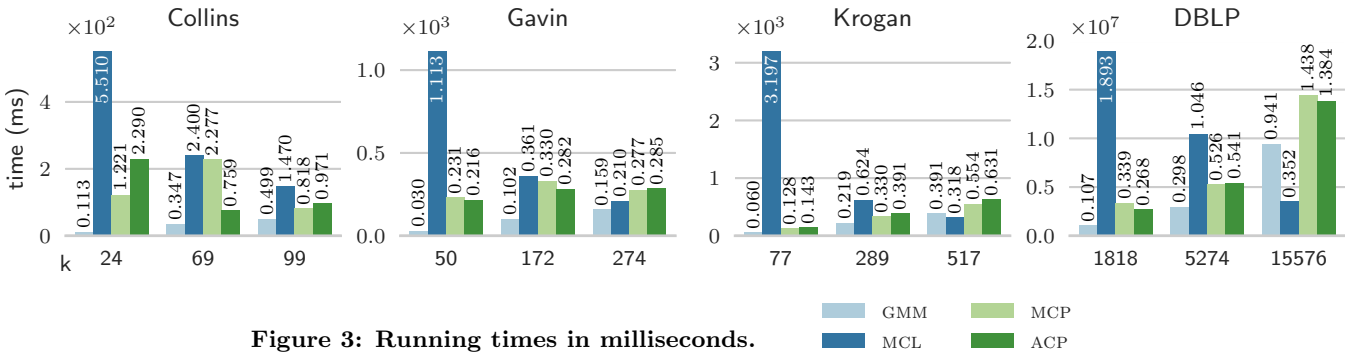

Figure 3: Running times in milliseconds.

the purpose of the evaluation, we restrict ourselves to proteins appearing in both `Krogan` and MIPS, thus obtaining a ground truth with 3,874 protein pairs. The input to the clustering algorithms is the entire `Krogan` graph. We evaluated the returned clusterings in terms of the confusion matrix. Namely, a pair of proteins assigned to the same cluster is considered a true positive if both proteins appear in the same MIPS complex, and a false positive otherwise.

For brevity, we restrict ourselves to exercise MCP and ACP considering only d-connection probabilities (see Section 3.4) for different values of d, and by setting $k = 547$ so as to match the cardinality of the reference clustering from [24]. The idea behind the use of limited path length is that we expect proteins of the same complex to be connected with high probability and, at the same time, to be topologically close in the graph. We do not report results for $d = 1$ since there is no clustering of the `Krogan` graph with $d = 1$ and $k = 547$. We compare the True Positive Rate (TPR) and the False Positive Rate (FPR) obtained by MCP with different values of d against those obtained with the MCL-based clustering of [24], and with the clustering computed by the algorithm in [23] (dubbed KPT in what follows). The results are shown in Table 2. We observe that, for small values of d, our algorithm is able to find a clustering with scores similar to the clustering of [24], while higher values of d yield fewer false negatives at the expense of an increased number of false positives. Note that ACP is slightly better than MCP w.r.t. the TPR, and MCP tends to be more conservative when it comes to the FPR. Furthermore, the FPR performance of ACP degrades more quickly as the depth increases. This is because ACP optimizes a measure that allows clusters where a few nodes are connected with low probability to their centers, and this effect amplifies as the depth increases. Thus, we can use MCP if we want to keep the number of false positives low, or ACP to achieve a higher TPR. Observe that a moderate number of false positives may be tolerable, since the corresponding protein pairs can be the target of further investigation to verify unknown protein interactions. Also, MCP, ACP, and MCL yield TPRs substantially higher than KTP[3]. This experiment supports our intuition that considering topologically close proteins, while aiming at high connection probabilities, makes our algorithm competitive with state-of-the-art solutions in the predictive setting.

[3]We remark that the performance of KTP reported here differs from the one reported in [23] since the ground truth considered in that paper only comprises pairs of proteins that appear in the same complex in the MIPS database and are connected by an edge in the `Krogan` graph, which clearly amplifies the TPR.

Table 2: MCP and ACP with limited path length against MCL and KPT on `Krogan` w.r.t. the MIPS ground truth.

		TPR		FPR	
		MCP	ACP	MCP	ACP
Depth	2	0.344	0.384	0.003	0.006
	3	0.416	0.459	0.012	0.078
	4	0.429	0.585	0.147	0.419
	6	0.695	0.697	0.604	0.633
	8	0.737	0.730	0.678	0.647
MCL [24]		0.423		0.002	
KPT [23]		0.187		$6.3 \cdot 10^{-4}$	

6. CONCLUSIONS

We presented a number of algorithms for center-based clustering of uncertain graphs. Unlike previous approaches, ours features provable guarantees on the clustering quality, and affords efficient implementations and an effective control on the number of clusters. We also provide an open source implementation of our algorithms which compares favorably with algorithms commonly used when dealing with uncertain graphs.

Our algorithms (MCP, ACP) consider objective functions capturing different statistics (minimum, average) of connection probabilities, that are not directly comparable. As such, there cannot be application-independent guidelines for choosing one of the two that apply to all practical scenarios. Considering that MCP and ACP feature comparable running times, the most reasonable approach could be to apply both and then choose the most "suitable" returned clustering, where suitability could be measured, for instance, through the use of external metrics such as inner/outer-AVPR.

Several challenges remain open for future research. From a complexity perspective, the conjectured NP-hardness of the ACP problem and, possibly, inapproximability results for both the MCP and ACP problems remain to be proved. More interestingly, there is still ample room for the development of practical algorithms featuring better analytical bounds on the approximation quality and/or faster performance, as well as for the investigation of other clustering problems on uncertain graphs.

7. ACKNOWLEDGMENTS

This work was supported by NSF grant IIS-1247581 and by University of Padova projects SID2017, CPDA152255.

8. REFERENCES

[1] E. Adar and C. Re. Managing uncertainty in social networks. *IEEE Data Engineering Bullettin*, 30(2):15–22, 2007.

[2] C. Aggarwal. *Managing and Mining Uncertain Data*. Advances in Database Systems. Springer US, 2010.

[3] S. Asthana, O. King, F. Gibbons, and F. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14(4):1170–1175, 2004.

[4] M. Ball. Computation compexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, R-35(3):230–239, 1986.

[5] O. Benjelloun, A. D. Sarma, A. Y. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *Proc. VLDB*, pages 953–964, 2006.

[6] A. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, 2005.

[7] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *PVLDB*, 5(11):1376–1387, 2012.

[8] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. SIGMOD*, pages 1247–1250. ACM, 2008.

[9] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *Proc. SODA*, pages 946–957. ACM-SIAM, 2014.

[10] M. Ceccarello, C. Fantozzi, A. Pietracaprina, G. Pucci, and F. Vandin. Clustering uncertain graphs. *CoRR*, abs/1612.06675, 2016.

[11] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. SODA*, pages 642–651. ACM-SIAM, 2001.

[12] S. Collins, P. Kemmeren, X. Zhao, J. Greenblatt, F. Spencer, F. C. Holstege, J. S. Weissman, and N. Krogan. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular & Cellular Proteomics*, 6(3):439–450, 2007.

[13] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, 2007.

[14] W. Ding. Extended most reliable source on an unreliable general network. In *Proc. ICICIS*, pages 529–533. IEEE, 2011.

[15] A. C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J.Jensen, S. Bastuck, B. Dümpelfeld, et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.

[16] J. Ghosh, H. Ngo, S. Yoon, and C. Qiao. On a routing problem within probabilistic graphs and its application to intermittently connected networks. In *Proc. INFOCOM*, pages 1721–1729. IEEE, 2007.

[17] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[18] Y. Gu, C. Gao, G. Cong, and G. Yu. Effective and efficient clustering methods for correlated probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1117–1130, 2014.

[19] D. Hochbaum and D. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[20] Institute of Bioinformatics and Systems Biology. The MIPS comprehensive yeast genome database. `ftp://ftpmips.gsf.de/fungi/Saccharomycetes/CYGD/`, May 2006.

[21] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. *PVLDB*, 4(9):551–562, 2011.

[22] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. KDD*, pages 137–146. ACM, 2003.

[23] G. Kollios, M. Potamias, and E. Terzi. Clustering large probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):325–336, 2013.

[24] N. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. Tikuisis, et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[25] L. Lin, J. Ruoming, C. Aggarwal, and S. Yelong. Reliable clustering on uncertain graphs. In *Proc. ICDM*, pages 459–468. IEEE, Dec 2012.

[26] H. Mewes, C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Münsterkötter, P. Pagel, N. Strack, V. Stümpflen, et al. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32(suppl 1):D41–D44, 2004.

[27] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[28] P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. Uncertain graph processing through representative instances. *ACM Transactions on Database Systems*, 40(3):20:1—20:39, 2015.

[29] A. Pietracaprina, M. Riondato, E. Upfal, and F. Vandin. Mining top-k frequent itemsets through progressive sampling. *Data Mining and Knowledge Discovery*, 21(2):310–326, 2010.

[30] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-nearest neighbors in uncertain graphs. *PVLDB*, 3(1):997–1008, 2010.

[31] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[32] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proc. SIGMOD*, pages 1539–1554. ACM, 2015.

[33] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[34] S. van Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[35] V. Vazirani. *Approximation Algorithms*. Springer, 2001.