Risk-Averse Matchings over Uncertain Graph Databases

Charalampos E. Tsourakakis[1], Shreyas Sekar[2], Johnson Lam[1], and Liu Yang[3]

[1] Boston University
ctsourak,jlam17@bu.edu
[2] University of Washington
sekarshr@uw.edu
[3] Yale University
liu.yang@yale.edu

Abstract. In this work we study a problem that naturally arises in the context of several important applications, such as online dating, kidney exchanges, and team formation.

> Given an uncertain, weighted (hyper)graph, how can we efficiently find a (hyper)matching with high expected reward, and low risk?

We introduce a novel formulation for finding matchings with maximum expected reward and bounded risk under a general model of uncertain weighted (hyper)graphs. Given that our optimization problem is NP-hard, we turn our attention to designing efficient approximation algorithms. For the case of uncertain weighted graphs, we provide a $\frac{1}{3}$-approximation algorithm, and a $\frac{1}{5}$-approximation algorithm with near optimal run time. For the case of uncertain weighted hypergraphs, we provide a $\Omega(\frac{1}{k})$-approximation algorithm, where k is the rank of the hypergraph (i.e., any hyperedge includes at most k nodes), that runs in almost (modulo log factors) linear time.

We complement our theoretical results by testing our approximation algorithms on a wide variety of synthetic experiments, where we observe in a controlled setting interesting findings on the trade-off between reward, and risk. We also provide an application of our formulation for providing recommendations of teams that are likely to collaborate, and have high impact. Our code is available at `https://github.com/tsourolampis`.

1 Introduction

Graphs model a wide variety of datasets that consist of a set of entities, and pairwise relationships among them. In several real-world applications, these relationships are inherently uncertain. For example, protein-protein interaction (PPI) networks are associated with uncertainty since protein interactions are obtained via noisy, error-prone measurements [4]. In privacy applications, deterministic edge weights become appropriately defined random variables [7,23], in dating applications each recommended link is associated with the probability that a date will be successful [11], in viral marketing the extent to which an idea propagates through a network depends on the 'influence probability' of each social interaction [24], in link prediction possible interactions are assigned probabilities [30,39], and in entity resolution a classifier outputs for each pair of entities a probability that they refer to the same object.

Mining uncertain graphs poses significant challenges. Simple queries—such as distance queries—on deterministic graphs become #**P**-complete ([42]) problems on uncertain graphs [19]. Furthermore, approaches that maximize the expected value of a given objective typically involve high risk solutions, e.g., solutions where there is an unacceptably large probability that the realized value of the objective is much smaller than its expected value. On the other hand, risk-averse methods are based on obtaining several graphs samples, a procedure that is computationally expensive, or even prohibitive for large-scale uncertain graphs.

Two remarks about the uncertain graph models used in prior work are worth making before we discuss the main focus of this work. The datasets used in the majority of prior work are *uncertain, unweighted graphs*. There appears to be less work related to *uncertain, weighted hypergraphs* that are able to model a wider variety of datasets, specifically those containing more than just pairwise relationships (i.e., hyperedges). Secondly, the model of uncertain graphs used in prior work [9,18,25,27,31,34,35,36] are homogeneous random graphs [8]. More formally, let $\mathcal{G} = (V, E, p)$ be an uncertain graph where $p : E \to (0, 1]$, is the function that assigns a probability of success to each edge independently from the other edges. According to the possible-world semantics [8,13] that interprets $\mathcal{G}$ as a set $\{G : (V, E_G)\}_{E_G \subseteq E}$ of $2^{|E|}$ possible deterministic graphs (worlds), each defined by a subset of E. The probability of observing any possible world $G(V, E_G) \in 2^E$ is

$$\mathbf{Pr}\left[G\right] = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e)).$$

Such a model restricts the distribution of each edge to be a Bernoulli distribution, and does not capture various important applications such as privacy applications where noise (say Gaussian) is injected on the weight of each edge [7,23].

In this work, we focus on *risk-averse matchings over uncertain (hyper)graphs*. To motivate our problem consider Figure 1 that shows a probabilistic graph (i.e., a 2-regular hypergraph) with two perfect matchings, $M_1 = \{(A, B), (C, D)\}$ and $M_2 = \{(A, C), (B, D)\}$. Each edge e follows a Bernoulli distribution with success probability $p(e)$, and is associated with a reward $w(e)$ that is obtained only when the edge is successfully realized. These two parameters $(p(e), w(e))$ annotate each edge e in Figure 1. The maximum weight matching *in expectation* is M_1 with expected reward $100 \times \frac{1}{2} \times 2 = 100$. However, with probability $(1-\frac{1}{2}) \times (1-\frac{1}{2}) = \frac{1}{4}$ the reward we receive from M_1 equals zero. However, the second matching M_2 has expected reward equal to 80 with probability 1. In other words, matching M_1 offers potentially higher reward but entails *higher risk* than M_2. Indeed, in many situations with asymmetric rewards, one observes that high reward solutions are accompanied by higher risks and that such solutions may be shunned by agents in favor of safer options [26].

Another way to observe that matching M_1 entails greater risk is to draw graph samples from this probabilistic graph multiple times, and observe that around 25% of the realizations of M_1 result in zero reward. However, sampling is computationally expensive on large-scale uncertain graphs. Furthermore, in order to obtain statistical guarantees, a large number of samples may be needed [35] which makes the approach computationally intensive or infeasible even for medium-scale graphs. Finally, it is challenging and sometimes not always clear how to aggregate different samples [35]. These two drawbacks are well-known to the database community, and recently Parchas et al. [35] suggested a heuristic to extract representative instances of uncertain graphs. While their work makes an important practical contribution, their method is an intuitive heuristic whose theoretical guarantees and worst-case running time are not well understood [35].

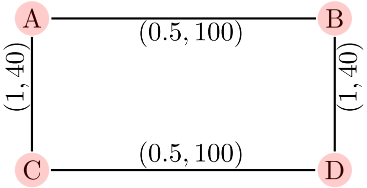


Fig. 1: Probabilistic graph, each edge e is annotated with $(p(e), w(e))$, its probability and its reward/weight. The matching $(A, B), (C, D)$ has higher expected weight than $(A, C), (B, D)$. However, the reward of the former matching is 0 with probability $\frac{1}{4}$, but the reward of the latter matching is 80 with probability 1. For details, see Section 1.

Motivated by these concerns, we focus on the following central question:

> How can we design *efficient*, *risk-averse algorithms* with *solid theoretical guarantees* for finding maximum weight matchings in uncertain weighted graphs and hypergraphs?

This question is well-motivated, as it naturally arises in several important applications. In online dating applications a classifier may output a probability distribution for the probability of matching two humans successfully [41]. In kidney exchange markets, a kidney exchange is successful according to some probability distribution that is determined by a series of medical tests. Typically, this distribution is unknown but its parameters such as the mean and the variance can be empirically estimated [11]. Finally, the success of any large organization that employs skilled human resources crucially depends on the choice of teams that will work on its various projects. Basic team formation algorithms output a set of teams (i.e., hyperedges) that combine a certain set of desired skills [3,17,20,29,33]. A classifier can leverage features that relate to crowd psychology, conformity, group-decision making, valued diversity, mutual trust, effective and participative leadership [22] to estimate the probability of success of a team.

In detail, our contributions are summarized as follows.

Novel Model and Formulation. We propose a general model for weighted uncertain (hyper)graphs, and a novel formulation for risk-averse maximum matchings. Our goal is to select (hyper)edges that have *high expected reward, but also bounded risk of failure.* Our problem is a novel variation of the well-studied stochastic matching problem [5,11].

Approximation algorithms. The problem that we study is NP-Hard so we focus on designing efficient approximation algorithms. For the case of uncertain graphs, using Edmond's blossom algorithm [15] as a black-box, we provide a risk-averse solution that is a $\frac{1}{3}$-approximation to the optimal risk-averse solution. Similarly, using a greedy matching algorithm as a black box we obtain a $\frac{1}{5}$-risk-averse approximation. For hypergraphs of rank k (i.e., any hyperedge contains at most k nodes) we obtain a risk-averse $\Omega(\frac{1}{k})$-approximation guarantee. Our algorithms are risk-averse, do not need to draw graph samples, and come with solid theoretical guarantees. Perhaps more importantly, the proposed algorithms that are based on greedy matchings have a running time of $O(m \log^2 m + n \log m)$, where n, m represent the number of nodes, and (hyper)edges in the uncertain (hyper)graph respectively—this makes the algorithm easy to deploy on large-scale real-world networks such as the one considered in our experiments (see Section 4).

Experimental evaluation. We evaluate our proposed algorithm on a wide variety of synthetic experiments, where we observe interesting findings on the trade-offs between reward and risk. There appears to be little (or even no) empirical work on *uncertain, weighted hypergraphs.* We use the Digital Bibliography and Library Project (DBLP) dataset to create a hypergraph where each node is an author, each hyperedge represents a team of co-authors for a paper, the probability of a hyperedge is the probability of collaboration estimated from historical data, and the weight of a hyperedge is its citation count. This uncertain hypergraph is particularly interesting as there exist edges with high reward (citations) but whose authors have low probability to collaborate. On the other hand, there exist papers with a decent number of citations whose co-authors consistently collaborate. Intuitively, the more risk-averse we are, the more we should prefer the latter hyperedges. We evaluate our proposed method on this real dataset, where we observe several interesting findings. The code and the datasets are publicly available at `https://github.com/tsourolampis/risk-averse-graph-matchings`.

2 Related Work

Uncertain graphs. Uncertain graphs naturally model various datasets including protein-protein interactions [4,28], kidney exchanges [37], dating applications [11], sensor networks whose connectivity links are uncertain due to various kinds of failures [38], entity resolution [34], viral marketing [24], and privacy-applications [7].

Given the increasing number of applications that involve uncertain graphs, researchers have put a lot of effort in developing algorithmic tools that tackle several important graph mining problems, see [9,18,25,27,31,34,35,36]. However, with a few exceptions these methods suffer from a critical drawback; either they are not risk-averse, or they rely on obtaining many graphs samples. Risk-aversion has been implicitly discussed by Lin et al. in their work on reliable clustering [31], where the authors show that interpreting probabilities as weights does not result in good clusterings. Jin et al. provide a risk-averse algorithm for distance queries on uncertain graphs [19]. Parchas et al. have proposed a heuristic to extract a good possible world in order to combine risk-aversion with efficiency [35]. However, their work comes with no guarantees.

Graph matching is a major topic in combinatorial optimization. The interested reader should confer the works of Lovász and Plummer [32] for a solid exposition. Finding maximum matchings in weighted graphs is solvable in polynomial time [15,16]. A faster algorithm sorts the edges by decreasing weight, and adds them to a matching greedily. This algorithm is a $\frac{1}{2}$-approximation to the optimum matching. Finding a maximum weight hypergraph matching is NP-hard, even in unweighted 3-uniform hypergraphs (a.k.a 3-dimensional matching) [21]. The greedy algorithm provides a $\frac{1}{k}$-approximation (intuitively for each hyperedge we greedily add to the matching, we lose at most k hyperedges) where k is the maximum cardinality of an edge.

Stochastic Matchings. Various stochastic versions of graph matchings have been studied in the literature. We discuss two papers that lie close to our work [5,11]. Both of these works consider a random graph model with a Bernoulli distribution on each edge. In contrast to our work, these models allow the central designer to *probe* each edge to verify its realization: if the edge exists, it gets irrevocably added to the matching. While Chen et al. [11] provide a constant factor approximation on unweighted graphs based on a simple greedy approach, Bansal et al. [5] obtain a $O(1)$-factor for even weighted graphs using an LP-rounding algorithm. On the other hand, our work focuses on designing fast algorithms that achieve good matchings with bounded risk on weighted graphs without probing the edges. Finally, since the hypergraph matching is also known as set packing, the above problems are special cases of stochastic set packing problem[14].

3 Model and Proposed Method

Uncertain Weighted Bernoulli hypergraphs. Before we define a general model for uncertain weighted hypergraphs that allows for both continuous and discrete probability distributions, we introduce a simple probabilistic model for weighted uncertain hypergraphs that generalizes the existing model for random graphs. Each edge e is distributed as a weighted Bernoulli variable independently from the rest: with probability $p(e)$ it exists, and its weight/reward is equal to $w(e)$, and with the remaining probability $1-p(e)$ it does not exist, i.e., its weight is zero. More formally, let $\mathcal{H} = ([n], E, p, w)$ be an uncertain hypergraph on n nodes with $|E| = m$ potential hyperedges, where $p : E \to (0, 1]$, is the function that assigns a probability of existence to each hyperedge independently from the other hyperedges, and $w : E \to \mathbb{R}^+$. The value $w(e)$ is the reward we receive from hyperedge e if it exists. Let $r_e \overset{\text{def}}{=} p(e)w(e)$ be the expected reward from edge e. According to the possible-world semantics [8,13], the probability of observing any possible world $H(V, E_H) \in 2^E$ is $\mathbf{Pr}[H] = \prod_{e \in E_H} p(e) \prod_{e \notin E_H} (1 - p(e))$.

Uncertain Weighted hypergraphs. More generally, let $\mathcal{H}([n], E, \{f_e(\theta_e)\}_{e \in E})$ be an uncertain hypergraph on n nodes, with hyperedge set E. The reward $w(e)$

of each hyperedge $e \in E$ is drawn according to some probability distribution f_e with parameters $\boldsymbol{\theta_e}$, i.e., $w(e) \sim f_e(x; \boldsymbol{\theta_e})$. We a wessume that the reward for each hyperedge is drawn independently from the rest; each probability distribution is assumed to have finite mean, and finite variance. Given this model, we define the probability of a given hypergraph H with weights $w(e)$ on the hyperedges as:

$$\mathbf{Pr}\left[H; \{w(e)\}_{e \in E}\right] = \prod_{e \in E} f_e(w(e); \boldsymbol{\theta_e}).$$

Our model allows for both discrete and continuous distributions, as well as mixed discrete and continuous distributions. In our experiments (Section 4) we focus on the weighted Bernoulli, and Gaussian cases.

Problem definition. Our goal is to output a matching M with high expected reward and low variance. A crucial assumption that we make is that for any given edge e, the algorithm designer does not have access to the complete distribution $f_e(\cdot)$ but only simple statistics such as its mean and standard deviation (s.t.d). Let $\mathcal{M}$ be the set of all matchings from the hyperedge set E. The total associated reward with a matching $M \in \mathcal{M}$ is the expected reward, i.e.,

$$R(M) \stackrel{\text{def}}{=} \sum_{e \in M} r_e = \sum_{e \in M} E_{f_e}[w(e)].$$

Similarly, the associated risk in terms of the standard deviation is defined as

$$risk(M) \stackrel{\text{def}}{=} \sum_{e \in M} \sigma_e,$$

where σ_e denotes the standard deviation of the distribution $f_e(x; \boldsymbol{\theta_e})$.

Given an uncertain weighted hypergraph, and a risk upper-bound B, our goal is to maximize the expected reward over all matchings with risk at most B. We refer to this problem as the Bounded Risk Maximum Weighted Matching (BR-MWM) problem. Specifically,

$$\boxed{\begin{array}{ll} \max\limits_{M \in \mathcal{M}} & R(M) \qquad \text{[BR-MWM problem]} \\ \text{s.t} & risk(M) \leq B \end{array} \qquad (1)}$$

For example, in the case of a weighted Bernoulli hypergraph where each hyperedge $e \in E$ exists with probability $p(e)$ and has weight $b(e)$ when it exists, formulation (1) becomes

$$\begin{array}{ll} \max\limits_{M \in \mathcal{M}} & \sum_{e \in M} p(e)b(e) \\ \text{s.t} & \sum\limits_{e \in M} b(e)\sqrt{p(e)(1-p(e))} \leq B \end{array} \qquad (2)$$

Similar formulations can be obtained for other specific distributions such as Gaussian. Finally, we remark that the BR-MWM problem is NP-Hard even on graphs via a simple reduction from Knapsack.

Other Measures of Risk. It is worth outlining that our model and proposed method adapts easily to other risk measures. For example, if we define the risk of a matching M in terms of its variance, i.e.,

$$risk(M) \overset{\text{def}}{=} \sum_{e \in M} \sigma_e^2, \tag{3}$$

then all of our theoretical guarantees and the insights gained via our experiments still hold with minor changes in the algorithm. At the end of this section, we discuss in detail the required changes. For the sake of convenience and concreteness, we present our results in terms of the standard deviation.

An LP-approximation algorithm. The BR-MWM problem is a special case of the *Hypermatching Assignment Problem* (HAP) introduced in [12]: given a k-uniform hypergraph $H(V, E)$, a budget B, a profit and a cost $w_e, c_e \geq 0$ for hyperedge e respectively, the goal of HAP is to compute a matching M so that the total profit $\sum_{e \in M} w_e$ is maximized and the budget constraint $\sum_{e \in M} c_e \leq B$ is satisfied. Therefore, we can invoke the randomized $\frac{1}{k+1+\epsilon}$-approximation algorithm for HAP [12] to solve our problem, here $\epsilon > 0$ is constant. However, this approach—at least for the moment—is unlikely to scale well: it requires solving a linear program with an exponential number of variables in terms of $\frac{1}{\epsilon}$, and then strengthen this LP by one round of the Lasserre's lift-and-project method. This motivates the design of scalable approximation algorithms.

3.1 Proposed Algorithm and Guarantees

Our algorithm is described in pseudocode 1. It takes as input a hypergraph matching algorithm Match-Alg as a black-box: the black-box takes a weighted hypergraph and returns a hypergraph matching. First, our algorithm removes all hyperedges e that have negative reward and for which $\sigma_e > B$ as they are not part of any optimal solution. For any given edge $e \in E$, define $\alpha_e \overset{\text{def}}{=} \frac{r_e}{\sigma_e}$. Now, we label the edges in E as $e_1, e_2, \ldots, e_m$ such that $\alpha_{e_1} \geq \alpha_{e_2} \geq \ldots \geq \alpha_{e_m}$, breaking ties arbitrarily. Sorting the α values requires $O(m \log m)$ time. Next, we consider the nested sequence of hypergraphs $\emptyset = H^{(0)} \subset H^{(1)} \subset \ldots \subset H^{(m)} = H$, where $H^{(i)}$ contains the i hyperedges $(e_1, e_2, \ldots, e_i)$, and each edge e is weighted by the expected reward r_e.

Let $M^{(i)}$ be the matching returned by Match-Alg on $H^{(i)}$ with weights $(r_e)_{e \in H^{(i)}}$. We first compute the maximum weight matching on $H^{(m)}$. If the quantity $risk(M^{(m)})$ is less than or equal to B, then we output $M^{(m)}$. Otherwise, we binary search the nested sequence of hypergraphs to find *any* index ℓ^* for which

$$risk(M^{(\ell^*)}) \leq B < risk(M^{(\ell^*+1)}).$$

The final output matching M_{OUT} is either $M^{(\ell^*)}$ or e_{ℓ^*+1}, depending on which one achieves greater expected reward. Intuitively, the latter case is required when there exists a single high-reward hyperedge whose risk is comparable to the upper bound B. In general, there may be more than one index ℓ^* that satisfies the above condition since the variance of $M^{(i)}$ is *not* monotonically increasing with i. Figure 2 provides such an example that shows that increasing the set of allowed edges can actually decrease the overall risk of the optimum matching.

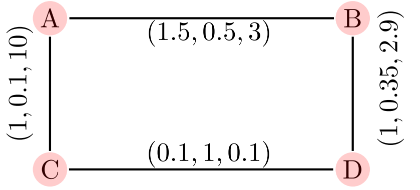


Fig. 2: The risk $risk(M^{(i)})$ of the optimum matching $M^{(i)}$ is *not* monotonically increasing with i. For details, see Section 3.

Specifically, Figure 2 shows an uncertain graph, each edge e is annotated with $(r_e, \sigma_e, \alpha_e)$. One can always find distributions that satisfy these parameters. We consider Algorithm 1 with the black-box matching algorithm Match-Alg as the optimum matching algorithm on weighted graphs. As our algorithm considers edges in decreasing order of their α-value, we get that $M^{(1)} = \{(A, C)\}, M^{(2)} = \{(A, B)\}, M^{(2)} = \{(A, B)\}, M^{(3)} = \{(A, C), (B, D)\}$. The risk of the above three matchings are $0.1, 0.5$, and 0.45 respectively. Thus, the quantity $risk(M^{(i)})$ is *not* monotonically increasing with i.

While it is not hard to see how a binary search would work, we provide the details for completeness. We know that $risk(M^{(1)}) = \sigma(e_1) \leq B$, and $risk(M^{(m)}) > B$. Let $low = 1, high = m$. We search the middle position mid between low and high, and $mid + 1$. If $risk(M^{(mid)}) \leq B < risk(M^{(mid+1)})$, then we set ℓ^* equal to mid and return. If not, then if $risk(M^{(mid)}) \leq B$, we repeat the same procedure with $low = mid + 1, high = m$. Otherwise, if $risk(M^{(mid)}) > B$ we repeat with $low = 1, high = mid$. This requires $O(\log m)$ iterations, and each iteration requires the computation of at most two matchings using the black-box Match-Alg.

Our proposed algorithm uses the notion of a black-box reduction: wherein, we take an arbitrary c-approximation algorithm for computing a maximum-weight hypermatching (Match-Alg, $c \leq 1$) and leverage its properties to derive an algorithm that in addition to maximizing the expected weight also has low risk. This black-box approach has a significant side-effect: organizations may have already invested in graph processing software for deterministic graphs can continue to use the same methods (as a black-box) regardless of the uncertainty inherent in the data. Our search takes time $O(\log m \times T(n, m))$ where $T(n, m)$ is the running time of maximum weighted matching algorithm Match-Alg.

$\frac{1}{3}$-approximation for uncertain weighted graphs. First we analyze our algorithm for the important case of uncertain weighted graphs. Unlike general hypergraphs, we can find a maximum weight graph matching in polynomial time using Edmond's algorithm [16]. Our main result is stated as the following theorem.

Theorem 1. *Assuming an exact maximum weight matching algorithm* Match-Alg*, Algorithm 1 returns a matching M_{OUT} whose risk is less than or equal to*

B, and whose expected reward is at least $\frac{1}{3}$ of the optimal solution to the Bounded Risk Maximum Weighted Matching problem on uncertain weighted graphs.

Before we prove Theorem 1, it is worth reiterating that our proposed algorithm provides a better approximation than the factor guaranteed in [12], i.e., $\frac{1}{3} > \frac{1}{3+\epsilon}$ for any constant $\epsilon > 0$. Additionally, our approach is orders of magnitude faster than the algorithm from [12] as the latter uses an LP-rounding technique, whereas our approach is simple and combinatorial.

Proof. Let M^{OPT} denote an optimum matching whose risk is at most B. Since it is immediately clear by the description of our algorithm that $risk(M_{OUT}) \leq B$, our goal is to prove that the matching returned by our algorithm has reward at least one-third as good as the reward of the optimum matching, i.e., $R(M_{OUT}) = \sum\limits_{e \in M_{OUT}} r_e \geq \frac{R(M^{OPT})}{3}$.

In order to show this bound, we prove a series of inequalities. By definition, $H^{(\ell^*+1)}$ differs from $H^{(\ell^*)}$ in exactly one edge, that is e_{ℓ^*+1}. We also know that the maximum weight matching in $H^{(\ell^*+1)}$ (i.e., $M^{(\ell^*+1)}$) is different from the maximum weight matching in $H^{(\ell^*)}$ ($M^{(\ell^*)}$) since the former entails risk that exceeds the budget B. We conclude that $M^{(\ell^*+1)}$ contains the edge e_{ℓ^*+1}.

Therefore, we have that $R(M^{(\ell^*+1)}) = R(M^{(\ell^*+1)} \backslash e_{\ell^*+1}) + r_{e_{\ell^*+1}} \leq R(M^{(\ell^*)}) + r_{e_{\ell^*+1}}$. This is true because $M^{(\ell^*)}$ is the maximum weight matching in $H^{(\ell^*)}$ and so its weight is larger than or equal to that of $M^{(\ell^*+1)} \setminus e_{\ell^*+1}$. In conclusion, our first non-trivial inequality is:

$$R(M^{(\ell^*)}) + r_{e_{\ell^*+1}} \geq R(M^{(\ell^*+1)}) \tag{4}$$

Next, we lower-bound $M^{(\ell^*+1)}$ by using the facts that $\alpha_e \geq \alpha_{e_{\ell^*+1}}$ for all $e \in M^{(\ell^*+1)}$, and that the total risk of $M^{(\ell^*+1)}$ is at least B by definition. Specifically,

$$\begin{aligned} R(M^{(\ell^*+1)}) &= \sum_{e \in M^{(\ell^*+1)}} r_e = \sum_{e \in M^{(\ell^*+1)}} \alpha_e \sigma_e && (5) \\ &\geq \sum_{e \in M^{(\ell^*+1)}} \alpha_{e_{\ell^*+1}} \sigma_e \\ &= \alpha_{e_{\ell^*+1}} \sum_{e \in M^{(\ell^*+1)}} \sigma_e > \alpha_{e_{\ell^*+1}} B. && (6) \end{aligned}$$

Now we show upper bounds on the optimum solution to the BR-MWM problem M^{OPT}. We divide M^{OPT} into two parts: M_1^{OPT} and M_2^{OPT}, where the first part is the set of edges in $M^{OPT} \cap H^{(\ell^*)}$ and the second part is the edges not present in $H^{(\ell^*)}$. We present separate upper bounds on M_1^{OPT} and M_2^{OPT}. By definition, M_1^{OPT} is a matching on the set of edges $H^{(\ell^*)}$. Therefore, its reward is smaller than or equal to that of the optimum matching on $H^{(\ell^*)}$, which happens to be $M^{(\ell^*)}$. Hence,

$$R(M_1^{OPT}) \leq R(M^{(\ell^*)}). \tag{7}$$

Algorithm 1 Algorithm for computing a $\frac{c}{2+c}$-approximate matching for the BR-MWM problem on uncertain weighted hypergraphs.

```
Require: $\mathcal{H}([n], E), (r_e = E_{f_e}[w_e])_{e \in E}, (\sigma_e = \sqrt{E_{f_e}[(w_e - r_e)^2]})_{e \in E}$, MATCH-ALG
  Remove all hyperedges $e$ that have either $r_e \leq 0$ or $\sigma_e > B$
  Sort hyperedges in decreasing order according to $\alpha_e = \frac{r_e}{\sigma_e}$, let $\alpha_{e_1} \geq \ldots \geq \alpha_{e_m} \geq 0$.
  $M^{(m)} \leftarrow$ MATCH-ALG$(H^{(m)})$
  if $risk(M^{(m)}) \leq B$ then
    $\ell^* \leftarrow m$
    Return $\ell^*, M^{(\ell^*)}$
  end if
  $low \leftarrow 1, high \leftarrow m$
  while True do
    $mid \leftarrow \lfloor \frac{low+high}{2} \rfloor$
    Compute $M^{(mid)}, M^{(mid+1)}$
    if $risk(M^{(mid)}) \leq B < risk(M^{(mid+1)})$ then
      $\ell^* \leftarrow mid$
      Return $\ell^*, M^{(\ell^*)}$
    else if $risk(M^{(mid)}) \leq B$ then
      $low \leftarrow mid + 1$
    else
      $high \leftarrow mid$
    end if
  end while
```

Next, consider M_2^{OPT}. To upper-bound $R(M_2^{OPT})$ we also use inequalities 4,6:

$$\begin{aligned} R(M_2^{OPT}) &= \sum_{e \in M_2^{OPT}} r_e = \sum_{e \in M_2^{OPT}} \alpha_e \sigma_e \\ &\leq \sum_{e \in M_2^{OPT}} \alpha_{e_{\ell^*+1}} \sigma_e = \alpha_{e_{\ell^*+1}} \sum_{e \in M_2^{OPT}} \sigma_e \\ &\leq \alpha_{e_{\ell^*+1}} B < R(M^{(\ell^*+1)}) \\ &\leq R(M^{(\ell^*)}) + r_{e_{\ell^*+1}}. \end{aligned}$$

Now, we are ready to complete the proof. Recall that the output of the algorithm M_{OUT} satisfies $R(M_{OUT}) = \max(R(M^{(\ell^*)}), r_{e_{\ell^*+1}})$. Combining the upper bounds for M_1^{OPT} and M_2^{OPT} yields the desired approximation factor of $\frac{1}{3}$:

$$\begin{aligned} R(M^{OPT}) &\leq R(M^{(\ell)}) + R(M^{(\ell)}) + r(e_{\ell+1}) \\ &= 2R(M^{(\ell)}) + r(e_{\ell+1}) \leq 3R(M_{OUT}). \blacksquare \end{aligned}$$

Running time: Assuming that the $O(mn + n^2 \log n)$ [16] implementation of Edmond's algorithm is used as a black-box, we remark that the run time of Algorithm 1 is $O(mn \log m + n^2 \log m \log n)$.

Fast $\frac{1}{5}$-approximation for uncertain weighted graphs. Since the running time using Edmond's algorithm is somewhat expensive, we show how the approximation guarantee changes when we use the (much faster) greedy algorithm for maximum weighted matchings as MATCH-ALG. Recall, the greedy matching algorithm runs in $O(m \log m + n)$ time.

Theorem 2. *If the black-box* MATCH-ALG *is set to be the greedy matching algorithm, then Algorithm 1 computes a $\frac{1}{5}$-approximation to the optimal solution of the BR-MWM problem in $O(m \log^2 m + n \log m)$-time.*

The proof is omitted as it is essentially identical to the proof of Theorem 1, with the only change that the greedy matching algorithm provides a $\frac{1}{2}$-approximation to the maximum weighted matching problem.

Fast $\frac{c}{2+c}$-approximation for uncertain weighted hypergraphs. Recall that finding a maximum weight hypergraph matching is NP-hard even for unweighted, 3-regular hypergraphs [21]. However, there exist various algorithms, that achieve different approximation factors $c < 1$. For example, the greedy algorithm provides a $\frac{1}{k}$-approximation guarantee, where k is the rank of the hypergraph (i.e., any hyperedge contains at most k nodes). Our main theoretical result follows.

Theorem 3. *Given any polynomial-time c-approximation algorithm* MATCH-ALG *($c \leq 1$) for the maximum weighted hypergraph matching problem, we can compute in polynomial time a hypermatching M_{OUT} such that its risk is at most B and its expected weight is a $\frac{c}{2+c}$-approximation to the expected weight of the optimal hypermatching that has risk at most B.*

Again the proof proceeds step by step as the proof of Theorem 1, and is omitted. In what follows, we restrict our attention to using the greedy hypermatching algorithm as a black-box. Our focus on greedy matchings stems from the fact that its approximation factor ($\frac{1}{k}$) is asymptotically optimal [6,10], that it is easy to implement, and runs in $O(m \log m + n)$ time using appropriate data structures. Since we will be using the greedy algorithm in our experiments (Section 4), we provide the following corollary.

Corollary 1. *For any hypergraph of rank k, we can compute in polynomial time a hypergraph-matching whose risk is at most B and whose weight is a $\Omega(\frac{1}{k})$ approximation to the optimum bounded-risk hypergraph matching.*

Algorithm 1 using the greedy hypermatching algorithm in lieu of MATCH-ALG runs in $O(m \log^2 m + n \log m)$ time.

Remark. We reiterate the point that our algorithm can be used to compute risk-averse matchings for other notions of risk such as variance. For instance, if we define risk as in Equation (3), then the only thing that changes in our algorithm is the definition of the α_e, namely that α_e is set equal to $\frac{r_e}{\sigma_e^2}$ for each (hyper)edge $e \in E$. The rest, including the theoretical guarantees remain identical.

4 Experimental Results

Experimental Setup and Normalization. We test our proposed algorithm on a diverse range of datasets, where the orders of magnitude of risk (e.g., standard deviation) can vary greatly across datasets. In order to have a consistent interpretation of the trade-off between expected reward and risk across datasets, we normalize the allowed risk B relative to the maximum possible standard deviation of a benchmark matching, $B_{\max}$. For the purpose of computing or more precisely approximating $B_{\max}$, we run the greedy matching algorithm on the (hyper)graph G (H) where the weight on edge e is σ_e, and set $B_{\max}$ to be the aggregate risk of the computed matching. While in theory one may observe a matching with greater risk than the obtained value $B_{\max}$, this does not occur in any of our simulations. We range B according to the rule $B = B_n \times B_{\max}$, where $B_n \in [0, 1]$ and is incremented in steps of 0.05. We refer to B_n as the *normalized risk* from now on. Due to space constraints we have not included a wide variety of synthetic experiments that can be found in an extended version of our work [40]. In the following we show our results on a real-world uncertain, weighted hypergraph. We implement our proposed fast approximation algorithm for uncertain weighted hypergraphs in Python. The code is available at Github [2]. All experiments were performed on a laptop with 1.7 GHz Intel Core i7 processor and 8GB of main memory.

Recommending impactful but probable collaborations. In many ways, academic collaboration is an ideal playground to explore the effect of risk-averse team formation for research projects as there exist teams of researchers that have the potential for high impact but may also collaborate less often. To explore this further, we use our proposed algorithm for uncertain weighted hypergraphs as a tool for identifying a set of disjoint collaborations that are both impactful and likely to take place. For this purpose, we use the Digital Bibliography and Library Project (DBLP) database. From each paper, we obtain a team that corresponds to the set of authors of that paper. As a proxy for the impact of the paper we use the citation count. Unfortunately, we could not obtain the citation counts from Google Scholar for the whole DBLP dataset as we would get rate limited by Google after making too many requests. Therefore, we used the AMiner citation network dataset [1] that contains citation counts, but unfortunately is not as up-to-date as Google Scholar is.

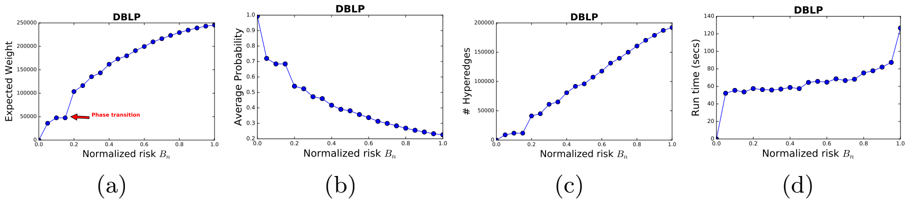
We preprocessed the dataset by removing all single-author papers since the corresponding hyperedge probabilities are one. Furthermore, multiple hyperedges are treated as one, with citation count equal to the sum of the citation counts of the multiple hyperedges. To give an example, if there exist three papers in the dataset that have been co-authored by authors A_1, A_2 with citation counts w_1, w_2, w_3 we create one hyperedge on the nodes that correspond to A_1, A_2 with weight equal $w_1 + w_2 + w_3$. If there exists another paper co-authored by A_1, A_2, A_3, this yields a different hyperedge/team $\{A_1, A_2, A_3\}$, and we do not include its citations in the impact of team $\{A_1, A_2\}$.

For hyperedge $e = (u_1, \ldots, u_\ell)$ we find the set of papers $\{P_1, \ldots, P_\ell\}$ authored by authors $u_1, \ldots, u_\ell$ respectively. We set the probability of hyperedge e as

$$p_e = \frac{|P_1 \cap P_2 \cap \ldots \cap P_\ell|}{|P_1 \cup P_2 \cup \ldots \cup P_\ell|}.$$

Intuitively, this is the empirical probability of collaboration between the specific set of authors.

To sum up, we create an uncertain weighted hypergraph using the DBLP dataset, where each node corresponds to an author, each hyperedge represents a paper whose reward follows a Bernoulli distribution with weight equal to the number of its citations, and probability p_e is the likelihood of collaboration. The final hypergraph consists of $n = 1,752,443$ nodes and $m = 3,227,380$ edges, and will be made publicly available on the first author's website. The largest collaboration involves a paper co-authored by 27 people, i.e., the rank k of the hypergraph is 27.



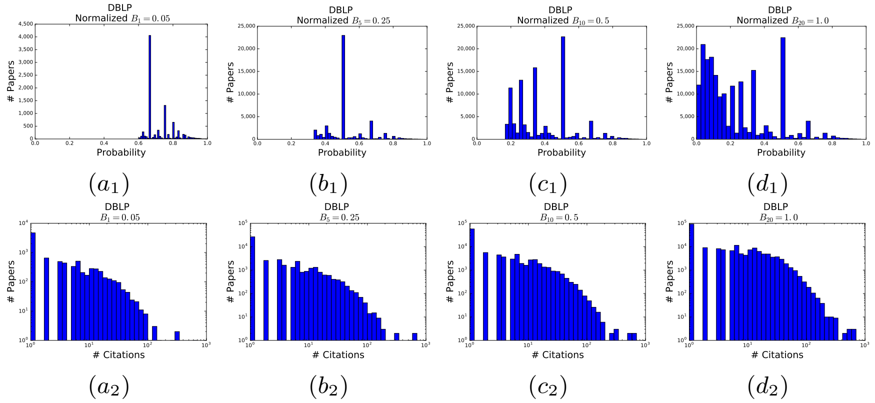
(a) (b) (c) (d)

Fig. 3: (a) Expected reward, (b) average probability (over hypermatching's edges), (c) number of edges in the hypermatching, and (d) running time in seconds versus normalized risk B_n. For details, see Section 4.

Figure 3 shows our findings when we vary the normalized risk bound B_n and obtain a hypermatching for each value of this parameter, using our algorithm. For the record, when $B_n = 1$, then $B = B_{\max} = 454\,392.0$. Figure 3(a) plots the expected weight of the hypermatching versus B_n. We observe an interesting phase transition when B_n changes from 0.15 to 0.2. This is because after $B_n = 0.15$ the average probability of the hyper-matching drops from ~ 0.7 to ~ 0.5. This is shown in Figure 3(b) that plots the average probability of the edges in each hypermatching computed by our algorithm vs. B_n. Figures 3(a),(b) strongly indicate what we verified by inspecting the output: up to $B_n = 0.15$, our algorithm picks teams of co-authors that tend to collaborate frequently. This finding illustrates that our tool may be used for certain anomaly detection tasks. Figures 3(c),(d) plot the number of hyperedges returned by our algorithm, and its running time in seconds vs B_n. We observe that a positive side-effect of using small risk bounds is speed: for small B_n values, the algorithm computes fewer maximum matchings.

By carefully inspecting the output of our algorithm for different B_n values, we see that at low values, e.g., $B_n = 0.05$, we find hyperedges typically with

50 to 150 citations with probabilities ranging typically from 0.66 to 1. When B_n becomes large we find hyper-edges with significantly more citations but with lower probability.



(a_1) (b_1) (c_1) (d_1)

(a_2) (b_2) (c_2) (d_2)

Fig. 4: Figures in first row $(a_1), (b_1), (c_1), (d_1)$ (second row $(a_2), (b_2), (c_2), (d_2)$): histograms showing the hyperedge probabilities (citations) in the hypermatching returned by our algorithm for normalized risk values B_n equal to $0.05, 0.25, 0.5, 1$ respectively. For details, see Section 4.

Finally, Figure 4 shows four pairs of histograms corresponding to the output of our algorithm for four different normalized risk values B_n, i.e., $0.05, 0.25, 0.5, 1$ respectively. Each pair $(\{(a_1),(a_2)\}$, $\{(b_1),(b_2)\}$, $\{(c_1),(c_2)\}$, and $\{(d_1),(d_2)\})$ plots the histogram of the probabilities, and the number of citations of the hyperedges selected by our algorithm for $B_n \in \{0.05, 0.25, 0.5, 1\}$ respectively. The histograms provide a view of how the probabilities decrease and citations increase as we increase B_n, i.e., as we allow higher risk.

5 Conclusion

In this work we study the problem of finding matchings with high expected reward and bounded risk on large-scale uncertain hypergraphs. We introduce a general model for uncertain weighted hypergraphs that allows for both continuous and discrete probability distributions, we provide a novel stochastic matching formulation that is NP-hard, and develop fast approximation algorithms. We verify the efficiency of our proposed methods on several synthetic and real-world datasets.

In contrast to the majority of prior work on uncertain graph databases, we show that it is possible to combine risk aversion, time efficiency, and theoretical guarantees simultaneously. Moving forward, a natural research direction is to design risk-averse algorithms for other graph mining tasks such as motif clustering, the k-clique densest subgraph problem, and k-core decompositions?

References

1. Aminer citation network dataset, August 2017. `https://aminer.org/citation`.
2. Risk-averse matchings over uncertain graph databases, January 2018. `https://github.com/tsourolampis/risk-averse-graph-matchings`.
3. A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proceedings of WWW 2012*, pages 839–848, 2012.
4. S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome research*, 14(6):1170–1175, 2004.
5. N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
6. P. Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. *Proceedings of SWAT 2000*, pages 31–40, 2000.
7. P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *Proceedings of the VLDB Endowment*, 5(11):1376–1387, 2012.
8. B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007.
9. F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In *Proceedings of the KDD 2014*, pages 1316–1325, 2014.
10. Y. H. Chan and L. C. Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Mathematical programming*, 135(1-2):123–148, 2012.
11. N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Proceedings of ICALP 2009*, pages 266–278. Springer, 2009.
12. M. Cygan, F. Grandoni, and M. Mastrolilli. How to sell hyperedges: The hypermatching assignment problem. In *Proceedings of SODA 2013*, pages 342–351, 2013.
13. N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.
14. B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of SODA 2005*, pages 395–404, 2005.
15. J. Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.
16. H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of SODA 1990*, pages 434–443, 1990.
17. A. Gajewar and A. Das Sarma. Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of ICDM 2012*, pages 165–176, 2012.
18. X. Huang, W. Lu, and L. V. Lakshmanan. Truss decomposition of probabilistic graphs: Semantics and algorithms. In *Proceedings of SIGMOD 2016*, pages 77–90, 2016.
19. R. Jin, L. Liu, and C. C. Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *Proceedings of KDD 2011*, pages 992–1000, 2011.
20. M. Kargar, A. An, and M. Zihayat. Efficient bi-objective team formation in social networks. *Machine Learning and Knowledge Discovery in Databases*, pages 483–498, 2012.
21. R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

22. J. R. Katzenbach. *Peak performance: Aligning the hearts and minds of your employees*. Harvard Business Press, 2000.
23. M. Kearns, A. Roth, Z. S. Wu, and G. Yaroslavtsev. Private algorithms for the protected in social network search. *Proceedings of the National Academy of Sciences*, 113(4):913–918, 2016.
24. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of KDD 2003*, pages 137–146. ACM, 2003.
25. A. Khan and L. Chen. On uncertain graphs modeling and queries. *Proceedings of the VLDB Endowment*, 8(12):2042–2043, 2015.
26. G. Kolata. Grant system leads cancer researchers to play it safe. *New York Times*, 24, 2009.
27. G. Kollios, M. Potamias, and E. Terzi. Clustering large probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):325–336, 2013.
28. N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637, 2006.
29. T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proceedings of KDD 2009*, pages 467–476. ACM, 2009.
30. D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.
31. L. Liu, R. Jin, C. Aggarwal, and Y. Shen. Reliable clustering on uncertain graphs. In *Proceedings of ICDM 2012*, pages 459–468. IEEE, 2012.
32. L. Lovász and M. D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.
33. A. Majumder, S. Datta, and K. Naidu. Capacitated team formation problem on social networks. In *Proceedings of KDD 2012*, pages 1005–1013, 2012.
34. W. E. Moustafa, A. Kimmig, A. Deshpande, and L. Getoor. Subgraph pattern matching over uncertain graphs with identity linkage uncertainty. In *Proceedings of ICDE 2014*, pages 904–915. IEEE, 2014.
35. P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. The pursuit of a good possible world: extracting representative instances of uncertain graphs. In *Proceedings SIGMOD 2014*, pages 967–978, 2014.
36. M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *Proceedings of the VLDB Endowment*, 3(1-2):997–1008, 2010.
37. A. E. Roth, T. Sönmez, and M. U. Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.
38. A. K. Saha and D. B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92. ACM, 2004.
39. C. E. Tsourakakis, M. Mitzenmacher, J. Błasiok, B. Lawson, P. Nakkiran, and V. Nakos. Predicting positive and negative links with noisy queries: Theory & practice. *arXiv preprint arXiv:1709.07308*, 2017.
40. C. E. Tsourakakis, S. Sekar, J. Lam, and L. Yang. Risk-averse matchings over uncertain graph databases. *arXiv preprint arXiv:1801.03190*, 2018.
41. K. Tu, B. Ribeiro, D. Jensen, D. Towsley, B. Liu, H. Jiang, and X. Wang. Online dating recommendations: matching markets and learning preferences. In *Proceedings of WWW 2014*, pages 787–792, 2014.
42. L. G. Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.