Commitment Under Uncertainty: Two-Stage Stochastic Matching Problems

Irit Katriel, Claire Kenyon-Mathieu and Eli Upfal
{irit,claire,eli}@cs.brown.edu

Brown University

January 25, 2007

Abstract

We define and study two versions of the bipartite matching problem in the framework of two-stage stochastic optimization with recourse. In one version the uncertainty is in the second stage costs of the edges, in the other version the uncertainty is in the set of vertices that needs to be matched. We prove lower bounds, and analyze efficient strategies for both cases. These problems model real-life stochastic integral planning problems such as commodity trading, reservation systems and scheduling under uncertainty.

Keywords: approximation algorithms, graph and network algorithms, randomized algorithms.

1 Introduction

Two-stage stochastic optimization with recourse is a popular model for hedging against uncertainty. Typically, part of the input to the problem is only known probabilistically in the first stage, when decisions have a low cost. In the second stage, the actual input is known but the costs of the decisions are higher. We then face a delicate tradeoff between speculating at a low cost vs. waiting for the uncertainty to be resolved.

This model has been studied extensively for problems that can be modeled by linear programming (LP) (sometimes using techniques such as Sample Average Approximation (SAA) when the LP is too large.) Recently there has been a growing interest in 2-stage stochastic combinatorial optimization problems [8, 14, 2, 25, 21, 5, 23, 22, 3]. Since an LP relaxation does not guarantee an integer solution in general, one can either try to find an efficient rounding technique [13] or develop a purely combinatorial approach [10, 7]. In order to develop successful algorithmic paradigms in this setting, there is an ongoing research program focusing on classical combinatorial optimization problems [24]: set cover, minimum spanning tree, Steiner tree, maximum weight matching, facility location, bin packing, multicommodity flow, minimum multicut, knapsack, and others. In this paper, we aim to enrich this research program by adding a basic combinatorial optimization problem to the list: the minimum cost maximum bipartite matching problem. The task is to buy edges of a bipartite graph which together contain a maximum-cardinality matching in the graph. We examine two variants of this problem. In the first, the uncertainty is in the second stage edge-costs, that is, the cost of an edge can either grow or shrink in the second stage. In the second variant, all edges

become more expensive in the second stage, but the set of nodes that need to be matched is not known.

Here are some features of minimum cost maximum bipartite matching that make this problem particularly interesting. First, it is not subadditive: the union of two feasible solutions is not necessarily a solution for the union of the two instances. In contrast, most previous work focused on subadditive structures, with the notable exception of Gupta and Pál's work on stochastic Steiner Tree [11]. Second, the solutions to two partial instances may interfere with one another in a way that seems to preclude the possibility of applying cost-sharing techniques associated with the scenario-sampling based algorithms [11, 12]. This intuitively makes the problem resistant to routine attempts, and indeed, we confirm this intuition by proving a lower bound which is stronger than what is known[1] for the sub-additive problems: in Theorem 5, we prove a hardness of approximation result in the setting where the second-stage scenarios are generated by choosing vertices independently and with identical probability. It is therefore natural that our algorithms yield upper bounds which are either rather weak (Theorem 2, Part 1) or quite specialized (Theorem 7). To address this issue, we relax the constraint that the output be a maximum matching, and consider bicriteria results, where there is a tradeoff between the cost of the edges bought and the size of the resulting matching (Theorem 2, Part 2, and Theorem 8). Such an approach may be a way to circumvent hardness for other stochastic optimization problems as well.

Although the primary focus of this work is stochastic optimization, another popular objective for the prudent investor is to minimize, not just the expected future cost, but the maximum future cost, over all possible future scenarios: that is the goal of robust optimization. We also prove a bicriteria result for robust optimization (Theorem 3.) Guarding oneself against the worst case is more delicate than just working with expectations. The solution requires a different idea: preventing undesirable high-variance events by explicitly deciding, against the advice of the LP solution, to not buy expensive edges (To analyze this, the proof of Theorem 3 involves some careful rounding.) This general idea might be applicable to other problems as well.

We note that within two-stage stochastic optimization with recourse, matching has been studied before [16]. However, the problem studied here is very different: there, the goal was to construct a maximum weight matching instead of the competing objective of large size and small cost; moreover the set of edges bought by the algorithm had to form a matching instead of just containing a matching. In the appendix, we give an example illustrating the difference between these two models.

Our main goal in this paper is to further fundamental understanding of the theory of stochastic optimization; however, we note that a conceivable application of this problem is commodity transactions, which can be viewed as a matching between supply and demand. When the commodity is indivisible, the set of possible transactions can be modeled as a weighted bipartite graph matching problem, where the weight of an edge represents the cost or profit of that transaction (including transportation cost when applicable). A trader tries to maximize profits or minimize total costs depending on her position in the transaction. A further tool that a commodity trader may employ to improve her income is timing the transaction. We model timing as a two-stage stochastic optimization problem with recourse: The trader can limit her risk by buying an option for a transaction at current information, or she can assume the risk and defer decisions to the second stage. Two common uncertainties in commodity transactions, price uncertainty and supply and demand un-

[1] To the best of our knowledge, all previous hardness results hold only when the second stage scenarios are given explicitly, i.e., when only certain combinations of parameter settings are possible.

certainty, correspond to the two stochastic two-stage matching problems mentioned above: finding minimum weight maximum matching with uncertain edge costs, and finding maximum matching with uncertain matching vertices. Similar decision scenarios involving matchings also show up in a variety of other applications such as scheduling and reservation systems.

Our results are summarized in the following table. We first prove (Theorem 1) that, with explicit scenarios, the uncertain matching vertices case is in fact a special case of the uncertain edge ws case. Then, it suffices to prove upper bounds for the more general variant and lower bounds for the restricted one. For the problem of minimizing the expected cost of the solution, we show an approximability lower bound of $\Omega(\log n)$. we then describe an algorithm that finds a maximum matching in the graph at a cost which is an n^2-approximation for the optimum. We then show that by relaxing the demand that the algorithm constructs a maximum matching, we can "beat" the lower bound: At a cost of at most $1/\beta$ times the optimum, we can match at least $n(1-\beta)$ vertices. Furthermore, we show that a similar bicriteria result holds also for the robust version of the problem, i.e., when we wish to minimize the worst-case cost incurred by the algorithm.

With independent choices in the second-stage scenarios, our main contribution is the lower bound. The reduction of Theorem 1 does not apply, but we prove APX-hardness for both types of uncertainty. We also prove an upper bound for a special case of the uncertain matching vertices variant.

Input:	Explicit Scenarios		Independent Choices
Criteria:	Expected Cost	Worst-Case Cost	Expected Cost
Uncertain edge costs	• n^2-approximation of the cost to get a maximum matching [Theorem 2, part 1] • $1/\beta$-approximation of the cost to match at least $n(1-\beta)$ vertices [Theorem 2, part 2] • Same hardness results as below [Theorem 1]	$1/\beta$-approximation of the cost to match at least $n(1-\beta)$ vertices [Theorem 3]	APX-hard [Theorem 6]
Uncertain matching vertices	• $\Omega(\log n)$ approximability lower bound [Theorem 4, Part 1] • NP-hard already for two scenarios [Theorem 4, Part 2] • Same upper bounds as above [Theorem 1]	same upper bound as above [Theorem 1]	• APX-hard [Theorem 5] • approximation for a special case [Theorem 7]

2 Explicit scenarios

In this section, we assume that we have an explicit list of possible scenarios for the second stage.

Uncertain edge costs. Given a bipartite graph $G = (A, B, E)$, we can buy edge e in the first stage at cost $C_e \geq 0$, or we can buy it in the second stage at cost $C_e^s \geq 0$ determined by the scenario s. The input has an explicit list of scenarios, and known edge costs (c_e^s) in scenario s. For uncertain edge costs, without loss of generality we can assume that $|A| = |B| = n$ and that G has a perfect matching. Indeed, there is an easy reduction from the case where the maximum matching has size

k: just create a new graph by adding a set A' of $n-k$ vertices on the left side, a set B' of $n-k$ vertices on the right side, and edges between all vertex pairs in $A' \times B$ and in $A \times B'$, with cost 0.

In the *stochastic optimization* setting, the algorithm also has a known second stage distribution: scenario s occurs with probability $\Pr(s)$. The goal is, in time polynomial in both the size of the graph and the number of scenarios, to minimize the *expected* cost; if E_1 denotes the set of edges bought in the first stage and E_2^s the set of edges bought in the second stage under scenario s, then:

$$\mathrm{OPT}_1 = \min_{E_1, E_2^s} \left\{ \sum_{s \in S} \Pr(s) \left(\sum_{e \in E_1} C_e + \sum_{e \in E_2^s} C_e^s \right) : \forall s, E_1 \cup E_2^s \text{ contains a perfect matching} \right\} \quad (1)$$

Stochastic optimization with uncertain edge costs has been studied for many problems, see for example [12, 19].

In the *robust optimization* setting, the goal is to minimize the maximum cost (instead of the expected cost):

$$\mathrm{OPT}_2 = \min_{E_1, E_2^s} \left\{ \max_{s \in S} \left(\sum_{e \in E_1} C_e + \sum_{e \in E_2^s} C_e^s. \right) : \forall s, E_1 \cup E_2^s \text{ contains a perfect matching} \right\} \quad (2)$$

Robust optimization with uncertain edge costs has also been studied for many problems, see for example [6].

Uncertain activated vertices. In this variant of the problem, there is a known distribution over scenarios s, each being defined by a set $B_s \subset B$ of *active* vertices that are allowed to be matched in that scenario. Each edge costs c_e today (before B_s is known) and τc_e tomorrow, where $\tau > 1$ is the *inflation parameter*. As in Expression 1, the goal is to minimize the expected cost, i.e.,

$$\mathrm{OPT}_3 = \left\{ C(E_1) + \tau \sum_{s \in S} \Pr(s) C(E_2^s) : \forall s, E_1 \cup E_2^s \text{ contains max matching of } (A, B_s, E \cap (A \times B_s)) \right\} \quad (3)$$

Stochastic optimization with uncertain activated vertices has also been previously studied for many problems, see for example [11]. There is a similar expression for robust optimization with uncertain activated vertices.

Theorem 1 (Reduction)**.** *The two-stage stochastic matching problem with uncertain activated vertices and explicit second-stage scenarios (OPT_3) reduces to the problem with uncertain edge costs and explicit second-stage scenarios (OPT_1).*

Proof. See appendix. □

From Theorem 1, it follows that our algorithms for uncertain edges costs (Theorems 2 and 3 below) imply corresponding algorithms for uncertain activated vertices as well, and that our lower bounds for uncertain activated vertices (Theorem 4 below) imply corresponding lower bounds for uncertain edge costs as well.

Theorem 2 (Stochastic optimization upper bound)**.** *1. There is a polynomial-time deterministic algorithm for stochastic matching (OPT_1) that returns a perfect matching whose overall expected cost is at most $2n^2 \cdot OPT_1$.*

2. *Given $\beta \in (0,1)$, there is a polynomial-time randomized algorithm for stochastic matching (OPT_1) that returns a matching whose cardinality, with probability $1-e^{-n}$ (over the random choices of the algorithm), is at least $(1-\beta)n$, and whose overall expected cost is $O(OPT_1/\beta)$.*

In particular, for any $\epsilon > 0$ we get a matching of size $(1-\epsilon)n$ and cost $O(\mathrm{OPT}/\epsilon)$ in expectation. Note that by Theorem 4, we have to relax the constraint on the size anyway if we wish to obtain a better-than-$\log n$ approximation on the cost, so Part 2 of the Theorem is, in a sense, our best option.

Proof. (Sketch). The proof follows the general paradigm applied to stochastic optimization in recent papers such as [13] for example: formulate the problem as an integer linear program; solve the linear relaxation and use it to guide the algorithm; and use linear programming duality (König's theorem, for our problem) for the analysis. To prove part 1, the algorithm buys, in the first stage, every edge whose associated LP variable is above a certain threshold; the analysis relies on Hall's theorem. To prove part 2, instead of a threshold the algorithm uses randomized rounding; the analysis relies on König's theorem. The detailed proof is in the Appendix. □

Theorem 3 (Robust optimization upper bound)**.** *Given $\beta \in (0,1)$, there is a polynomial-time randomized algorithm for robust matching (OPT_2) that returns a matching such that with probability at least $1-2/n$ (over the random choices of the algorithm), the following holds: In every scenario, the algorithm incurs cost $O(OPT_2(1+\ln(t)/\ln(n))/\beta)$ and outputs a matching of cardinality at least $(1-\beta)n$.*

Proof. We detail this proof, which is the most interesting one in this section. The integer programming formulation is similar to the one used to prove Theorem 2. More specifically, let X_e indicate whether edge e is bought in the first stage, and for each scenario s, let Z_e^s (resp. Y_e^s) indicate whether edge e is bought in the first stage (resp. in the second stage) and ends up in the perfect matching when scenario s materializes. We obtain:

$$\min W \text{ s.t. } \begin{cases} \sum_{e:v\in e}(Z_e^s + Y_e^s) = 1 & \forall v \in A \cup B \text{ and } \forall s \in S \\ Z_e^s \le X_e & \forall e \in E \text{ and } s \in S \\ \sum_e [C_e X_e + C_e^s Y_e^s] \le W & \forall s \in S \\ X_e, Y_e^s, Z_e^s \in \{0,1\} & \forall e \in E \text{ and } s \in S. \end{cases}$$

The algorithm solves the standard linear programming relaxation, in which the last set of constraints is replaced by $0 \le X_e, Y_e^s, Z_e^s \le 1$. Let $w, (x_e), (y_e^s), (z_e^s)$ denote the optimal solution of the linear program. Let $\alpha = 8\ln(2)/\beta$ again, and let $T = 3\ln n$.

- In the first stage, relabel the remaining edges so that $c_1 \ge c_2 \ge \cdots$. Let t_1 be maximum such that $x_1 + x_2 + \cdots + x_{t_1} \le T$. For every $j > t_1$, buy edge j with probability $1 - e^{-x_j\alpha}$. (Do not buy any edge $j \le t_1$.)
- In the second stage, relabel the remaining edges so that $c_1^s \ge c_2^s \ge \cdots$. Let t_2 be maximum such that $y_1^s + y_2^s + \cdots + y_{t_1}^s \le T$. For every $j > t_2$, buy edge j with probability $1 - e^{-y_j^s\alpha}$. (Do not buy any edge $j \le t_2$.)

Finally, the algorithm computes and returns a maximum matching of the set of edges bought.

We note that this construction and the rounding used in the analysis are almost identical to the construction used in strip-packing [15]. The analysis of the cost of the edges bought is the difficult

part. We first do a slight change of notations. The cost can be expressed as the sum of at most $2m$ random variables (at most m in each stage). Let $a_1 \geq a_2 \geq \cdots$ be the multiset $\{c_e\} \cup \{c_e^s\}$, along with the corresponding probabilities p_i ($p_i = 1 - e^{-x_e\alpha}$ if $a_i = c_e$ is a first-stage cost, and $p_i = 1 - e^{-y_e^s\alpha}$ if $a_i = c_e^s$ is a second-stage cost.) Let X_i be the binary variable with expectation p_i. Clearly, the cost incurred by the algorithm can be bounded above by $X = \sum_{i>t^*} a_i X_i$, where t^* is maximum such that $p_1 + \cdots + p_{T^*} \leq T$.

To prove a high-probability bound on X, we will partition $[1, 2m]$ into intervals to define groups. The first group is just $[1, t]$, and the subsequent groups are defined in greedy fashion, with group $[j, \ell]$ defined by choosing ℓ maximum so that $\sum_{i\in[j,\ell]} p_i \leq T$. Let $G_1, G_2, \ldots, G_r$ be the resulting groups. We have:

$$X \leq \sum_{\ell\geq 2}\sum_{i\in G_\ell} a_i X_i \leq \sum_{\ell\geq 2}\sum_{i\in G_\ell} (\max_{G_\ell} a_i) X_i \leq \sum_{\ell\geq 2}\sum_{i\in G_\ell} (\min_{G_{\ell-1}} a_i) X_i \leq \sum_{\ell\geq 1} (\min_{G_\ell} a_i) \sum_{i\in G_{\ell+1}} X_i.$$

On the other hand, (using the inequality $1 - e^{-Z} \leq Z$), the optimal value OPT^* of the linear programming relaxation satisfies:

$$\alpha\mathrm{OPT}^* \geq \sum_i a_i p_i \geq \sum_{\ell\geq 1}\sum_{i\in G_\ell} (\min_{G_\ell} a_i) p_i \geq \sum_{\ell\geq 1} (\min_{G_\ell} a_i)(T-1).$$

It only remains, for each group G_ℓ, to apply a standard Chernoff bound to bound the sum of the X_i's in G_ℓ, and use union bounds to put these results together and yield the statement of the theorem (see appendix.) □

We note that the proof of Theorem 3 can also be extended to the setting of Theorem 2 to prove a high probability result: For scenario s, with probability at least $1 - 2/n$ over the random choices of the algorithm, the algorithm incurs cost $O(\mathrm{OPT}_s/\beta)$ and outputs a matching of cardinality at least $(1-\beta)n$, where $\mathrm{OPT}_s = \sum_{E_1} C_e + \sum_{E_2^s} C_e^s$.
Finally, we can show two hardness of approximation results for the explicit scenario case.

Theorem 4 (Stochastic optimization lower bound)**.** *1. There exists a constant $c > 0$ such that Expression OPT_3 (Eq (3)) is NP-hard to approximate within a factor of $c \ln n$.*

2. Expression OPT_3 (Eq (3)) is NP-hard to compute, even when there are only two scenarios and τ is bounded.

Proof. The proof is in the appendix. The first part is proved by reduction from Minimum-Set-Cover [1] and the second is by reduction from the Simultaneous Matchings [9] problem. □

3 Implicit scenarios

Instead of having an explicit list of scenarios for the second stage, it is common to have instead an implicit description: in the case of uncertain activated vertices, a natural stochastic model is the one in which each vertex is active in the second stage with some probability p, independently of the status of the other nodes. Due to independence, we get that although the total number of possible scenarios can be exponentially large, there is a succinct description consisting of simply specifying the activation probability of each node. In this case, we can no longer be certain that the second-stage graph contains a perfect matching even if the input graph does, so the requirement is, as stated above, to find the largest possible matching.

We first prove an interesting lower bound.

3.1 Lower bounds

Theorem 5. *Stochastic optimization with uncertain vertex set is APX hard, even with independent vertex activation and identical activation probabilities.*

Proof. We detail this proof, which is the most interesting of our lower bounds.

We will use a reduction from Minimum 3-Set-Cover(2), the special case of Minimum Set-Cover where each set has cardinality exactly 3 and each element belongs to exactly two sets [18]. This variant is APX-hard and, in particular, it is NP-hard to approximate it to within a factor of 100/99 [4].

We will prove that approximating Expression (3) to within a factor of β is at least as hard as approximating 3-set-cover(2) to within a factor of $\gamma = \beta(1 + (3p^2(1-p) + 2p^3)\tau)$. APX-hardness follows by setting p to be a constant in the interval $[0, 0.0033]$ and $\tau = 1/p$, because then $3p^2(1-p) + 2p^3 > 1/99$.

Given an instance $(S = \{s_1, \ldots, s_n\}; C = \{c_1, \ldots, c_k\})$ of 3-set-cover(2), we construct an instance of the two-stage matching problem with uncertain activated vertices as follows (see Figure 1). The graph contains $2|S| + 3|C|$ vertices: for every element $s_i \in S$ there are two vertices u_i, u_i' connected by an edge whose first stage cost is 1; for every set $c_j \in C$, there are three vertices x_j, y_j, and z_j connected by a path $(x_j, y_j), (y_j, z_j)$. For every set c_j and element s_i which belongs to c_j, we have the edge (z_j, u_i). It is easy to see that the graph is bipartite. The first-stage edge costs are 1 for an (x_i, y_i) edge and 0 for the other edges. The second-stage costs are equal to the first-stage costs, multiplied by τ. In the second-stage scenarios, each vertex u_i is active with probability p.

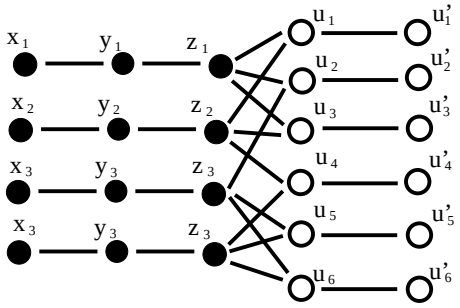

Figure 1: The graph obtained from the 3-Set-Cover(2) instance $\{s_1, s_2, s_3\}$, $\{s_1, s_3, s_4\}$, $\{s_2, s_5, s_6\}$, $\{s_4, s_5, s_6\}$.

If $p > 1/\tau$, then buying all (u_i, u_i') edges in the first stage at cost n is optimal. To see why, assume that an algorithm spends than $n' < n$ in the first stage. In the second stage, the expected number of active vertices that cannot be matched is at least $(n - n')p$ and the expected cost of matching them is $\tau(n - n')p < (n - n')$. We will assume in the following that $p \leq 1/\tau$.

Consider a minimum set cover $\mathcal{SC}$ of the input instance. Assume that in the first stage we buy (at cost 1) the edge (x_j, y_j) for every set $c_j \in \mathcal{SC}$. In the second stage, let I be the set of active vertices and find, in a way to be described shortly, a matching M_I between a subset I' of I and the vertex-set $\{z_j : c_j \in \mathcal{SC}\}$, using (z_j, u_i)-edges from the graph. Buy the edges in M_I (at cost 0). For every $i \in I \setminus I'$, buy the edge (u_i, u_i') at cost τ. Now, all active u_i vertices are matched, and it remains to ensure that the y-vertices are matched as well. Assume that y_j is unmatched. If z_j

is matched with some u_i node, this is because $c_j \in \mathcal{SC}$, so we bought the edge (x_j, y_j) in the first stage and can now use it at no additional cost. Otherwise, we buy the edge (y_j, z_j) at cost 0. The second stage has cost equal to τ times the cardinality of $I \setminus I'$ and the first stage has cost equal to the cardinality of the set cover.

The matching M_I is found in a straightforward manner: Given $\mathcal{SC}$, each element chooses exactly one set among the sets covering it, and, if it turns out to be active, will only try to be matched to that set. Each set in the set cover will be matched with one element, chosen arbitrarily among the active vertices who try to be matched with it. This defines the matching.

To calculate the expected cost of matching the vertices of $I - I'$, consider a set in $\mathcal{SC}$. It has 3 elements, and is chosen by at most 3 of them. Assume that it is chosen by all 3. With probability $(1-p)^3 + 3p(1-p)^2$, at most one of them is active and no cost is incurred in the second stage. With probability $3p^2(1-p)$, two of them are active and a cost of τ is incurred. With probability p^3, all three of them are active and a cost of 2τ is incurred, for an expected cost of $(3p^2(1-p) + 2p^3)\tau$. If the set is chosen by two elements, the expected cost is at most $p^2\tau$, and if it is chosen by fewer, the expected cost is 0. Thus in all cases the expected cost of matching $I \setminus I'$ is bounded by $|\mathcal{SC}|(3p^2(1-p) + 2p^3)\tau$. With a cost of $|\mathcal{SC}|$ for the first stage, we get that the total cost of the solution is at most $|\mathcal{SC}|(1 + (3p^2(1-p) + 2p^3)\tau)$.

On the other hand, let $\mathcal{M}_1$ be the set of cost-1 edges bought in the first stage. Let an (x_i, y_i) edge represent the set c_i and let a (u_i, u_i') edge represent the singleton set $\{s_i\}$. Now, assume that $\mathcal{M}_1$ does not correspond to a set cover of the input instance. Let x be the number of elements which are not covered by the sets corresponding to $\mathcal{M}_1$ and let X be the number of active elements among those x. In the second stage, the algorithm will have to match each uncovered element vertex u_i, either by its (u_i, u_i') edge (at cost n) or by a (z_j, u_i) edge for some set c_j where $s_i \in c_j$. In the latter case, if would have to buy the edge (x_i, y_i), again at cost n. The second stage cost, therefore, is at least Xn. But the expected value of X is x/n, thus the total expected cost is at least $|\mathcal{M}_1| + x$. Since we could complete $\mathcal{M}_1$ into a set cover by adding at most one set per uncovered element, we have $x + |\mathcal{M}_1| \geq |\mathcal{SC}|$.
In summary, we get that Expression (3) satisfies

$$|\mathcal{SC}| \leq \text{OPT} \leq |\mathcal{SC}|(1 + (3p^2(1-p) + 2p^2)\tau).$$

This means that if we can approximate our problem within a factor of β, then we can approximate Minimum 3-Set-Cover(2) within a factor of $\gamma = \beta(1 + (3p^2(1-p) + 2p^3)\tau)$, and the theorem follows. □

Using similar ideas we can also prove the following related result.

Theorem 6. *Stochastic optimization with uncertain, independent, edge costs is APX-hard, even with identical edge cost distributions.*

Proof. See appendix. □

3.2 Upper bound in a special case

We show that when $c_e = 1$ for all $e \in E$, it is possible to construct a perfect matching cheaply when the graph has certain properties. We study the case in which B is significantly larger than A.

Theorem 7. *Assume that the graph contains n vertex-disjoint stars $s_1, \ldots, s_n$ such that star s_i is centered at some vertex of A and contains $d = \max\{1, \ln(\tau p)\}/\ln(1/(1-p)) + 1$ vertices from B. Then there is an algorithm whose running time is polynomial in n and which returns a maximum-cardinality matching of the second stage graph, whose expected cost is $O(OPT_3 \cdot \min\{1, \ln(\tau p)\})$.*

To prove this, let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$. Let E_1 be the edges in the stars. Let B_2 be the vertices which are active in the second stage. Here is the algorithm. In the first stage, if $\tau p \leq e$ then the algorithm buys nothing; else, the algorithm buys all edges of E_1, paying nd. In the second stage, the algorithm completes its set of edges into a perfect matching in the cheapest way possible.

To analyze the algorithm, we say that a_i is *miserable* if none of the vertices in s_i are active and that it is *poor* if exactly one vertex in s_i is active. Let A_m be the set of miserable vertices and A_p the set of poor vertices. The following Lemma is the key of the analysis to constructing a perfect matching, and so we give its proof in detail.

Lemma 1. *There exists a maximum-cardinality matching M^* in G_2 such that*

$$|M^* \setminus E_1| \leq 2|A_m| + |A_p|.$$

Proof. Let M^* be a maximum matching in G_2 that has the maximum number of edges from E_1. Let M be a maximum matching that uses only edges from E_1. The edge-set $M \oplus M^*$ is a collection of vertex-disjoint odd-length paths, each of which connects a vertex a_i of A with a vertex b_j of B and is denoted $P(a_i, b_j)$; both a_i and b_j are unmatched in M. Since vertex a_i is unmatched in M, it must be that it is miserable. For each other vertex $a_k \in A \cap P(a_i, b_j)$, let (a_k, b_k) be the matching edge in M and (a_k, b_{k+1}) be the matching edge in M^*. If a_k is not poor, then there is another vertex $b_{k'}$ in the star centered on a_k, which is active but not matched in M. If $b_{k'}$ is not matched in M^*, then $(M^* \setminus \{(a_k, b_{k+1})\}) \cup \{(a_k, b_{k'})\}$ would be another maximum matching in G_2 with one more edge from E_1, contradicting the definition of M^*. Thus $b_{k'}$ is matched in M^*, but not in M. Let $P(a_{i'}, b_{k'})$ be the path of $M \oplus M^*$ that $b_{k'}$ belongs to: $a_{i'}$ is miserable. In this way, we can associate every rich A-vertex that lies on an alternating path with a unique miserable node. We get that the total number of vertices of A which are along the paths of $M \oplus M^*$ is at most $|A_m| + |A_p| + |A_m|$, hence the lemma.

Figure 2 illustrates the proof. It shows an alternating path starting at the unmatched, miserable vertex $a \in A$ to a rich vertex $d \in A$. For every rich A-vertex along the path (except for the last), such as the vertex b in the example, there is another alternating path that ends in this node's star. Hence, we can charge the miserable vertex at the head of that path (f in the example) for this rich internal node.

□

The following two lemmas are not difficult.

Lemma 2. *If $\tau p \geq e$ then the expected number of miserable vertices is $E|A_m| = n(1-p)/(\tau p)$ and the expected number of poor vertices is $E|A_p| = nd/\tau$. Otherwise, the expected number of miserable vertices is $E(|A_m|) = n(1-p)/e$.*

Proof. see appendix. □

Lemma 3. *The optimal cost is at least $(n - E(|A_m|)) \min(\tau, 1/p)$.*

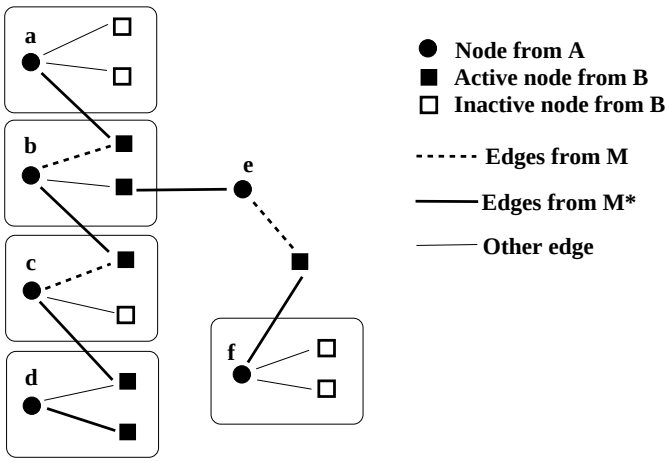

Figure 2: Illustration of the proof of Lemma 1.

Proof. see appendix. □

The rest of the proof is in the Appendix.

3.3 Generalization: The Black Box Model

With independently activated vertices, the number of scenarios is extremely large, and so solving a linear program of the kind described in previous sections is prohibitively time-consuming. However, in such a situation there is often a *black box* sampling procedure that provides, in polynomial time, an unbiased sample of scenarios; then one can use the SAA method to simulate the explicit scenarios case, and, if the edge cost distributions have bounded second moment, one can extend the analysis so as to obtain a similar approximation guarantee. The main observation is that the value of the LP defined by taking a polynomial number of samples of scenarios tightly approximates the the value of the LP defined by taking all possible scenarios. Using an analysis similar to [7] we can prove

Theorem 8. *Consider a two-stage edge stochastic matching problem with (1) a polynomial time unbiased sampling procedure and (2) edge cost distributions have bounded second moment. For any constants $\epsilon > 0$ and $\delta, \beta \in (0, 1)$, there is a polynomial-time randomized algorithm that outputs a matching whose cardinality is at least $(1-\beta)n$ and, with probability at least $1-\delta$ (over the choices of the black box and of the algorithm), incurs expected cost $O(OPT/\beta)$ (where the expectation is over the space of scenarios).*

Proof. Ommitted. □

References

[1] N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177, 2006.

[2] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

[3] Moses Charikar, Chandra Chekuri, and M. Pal. Sampling bounds fpr stochastic optimization. In *APPROX-RANDOM*, pages 257–269, 2005.

[4] M. Chlebík and J. Chlebíková. Inapproximability results for bounded variants of optimization problems. In *FCT 2003*, volume 2751 of *LNCS*, pages 27–38, 2003.

[5] D.B.Shmoys and C. Swamy. Stochastic optimization is almost as easy as deterministic optimization. In *45th IEEE FOCS*, pages 228–237, 2004.

[6] K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 367–378, Washington, DC, USA, 2005. IEEE Computer Society.

[7] K. Dhamdhere, R. Ravi, and M. Singh. On two-stage stochastic minimum spanning trees. In *IPCO*, volume 3509 of *Lecture Notes in Computer Science*, pages 321–334. Springer, 2005.

[8] S. Dye, L. Stougie, and A. Tomasgard. The stochastic single resource service-provision problem. *Naval Research Logistics*, 50:257–269, 2003.

[9] K. M. Elbassioni, I. Katriel, M. Kutz, and M. Mahajan. Simultaneous matchings. In *Algorithms and Computation, 16th International Symposium (ISAAC 2005)*, volume 3827 of *Lecture Notes in Computer Science*, pages 106–115. Springer, 2005.

[10] A. D. Flaxman, A. M. Frieze, and M. Krivelevich. On the random 2-stage minimum spanning tree. In *SODA*, pages 919–926. SIAM, 2005.

[11] A. Gupta and M. Pál. Stochastic steiner trees without a root. In *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1051–1063. Springer, 2005.

[12] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *STOC*, pages 417–426. ACM, 2004.

[13] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *FOCS*, pages 218–227. IEEE Computer Society, 2004.

[14] N. Immorlica, D. Karger, M. Minkoff, and V.S.Mirrokni. On the costs and benefits of procratination: approximation algorithms for stochastic combinatorial optimization problems. In *16th ACM-SIAM SODA*, pages 691–700, 2004.

[15] C. Kenyon and E. Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Math. Oper. Res.*, 25(4):645–656, 2000.

[16] N. Kong and A. J. Schaefer. A factor 1/2 approximation algorithm for two-stage stochastic matching problems. *European Journal of Operational Research*, 172:740–746, 2006.

[17] E. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, Winston, 1976.

[18] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computing and System Sciences*, 43:425–440, 1991.

[19] R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *IPCO*, volume 3064 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2004.

[20] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM Press, 1997.

[21] D. B. Shmoys and C. Swamy. The sample average approximation method for 2-stage stochastic optimization, 2004.

[22] David Shmoys and Mauro Sozio. Approximation algorithms for 2-stage stochastic scheduling problems. In *IPCO*, 2007.

[23] C. Swamy and D.B.Shmoys. The sampling-based approximation algorithms for multi-stage stochastic optimization. In *46th IEEE FOCS*, pages 357–366, 2005.

[24] Chaitanya Swamy and David B. Shmoys. Algorithms column: Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37(1):33–46, March 2006.

[25] B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24:289–333, 2003.

Appendix

The gap between Kong-Schaefer's stochastic matching model and our model

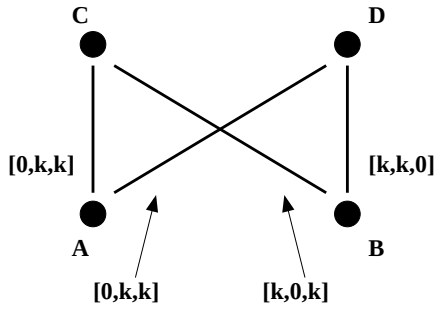

Figure 3: An example in which buying edges speculatively can help.

Kong and Shaefer [16] considered the two-stage stochastic matching problem with uncertain edge-costs, where the edges bought in the first stage must belong to the final matching constructed by the algorithm. The example in Figure 3 shows that cost of buying a matching in their model can be arbitrarily larger than the model in which speculative first-stage purchases are allowed (i.e., we are not required to use all edges bought in the first stage). There are two second-stage scenarios, each of which occurs with probability 1/2. For each edge, a vector $[c_e^0, c_e^1, c_e^2]$ of three edge costs is specified: c_e^0 is the first stage cost of the edge, c_e^1 is the second-stage cost in scenario 1 and c_e^2 is the second-stage cost in scenario 2. An optimal speculative algorithm buys both edges incident to A in the first stage at cost 0, and in the second stage buys the cost-0 edge incident to B. A non-speculative algorithm, on the other hand, can only buy one of the edges incident to A. If it does so, its expected cost in the second stage would be $k/2$. The other two options are worse: buying two edges in the first stage or buying to edges in the second stage costs k.

Proof of Theorem 1

We give an approximation preserving reduction from the stochastic matching vertices case.

Given an instance with stochastic matching vertices, we transform it to an instance of the problem with stochastic edge-costs, as follows. Assume that our input graph is $G = (A, B, E)$ where $A = \{a_1, \dots a_{|A|}\}$ and $B = \{b_1, \dots b_{|B|}\}$. We first add a set $A' = a'_1, \dots, a'_{|B|}$ of $|B|$ new vertices to A, and connect each a'_i with b_i by an edge. In other words, we generate the graph $G' = (A \cup A', B, E \cup \{(a'_i, b_i) : 1 \leq i \leq |B|\})$.

For the edges between A and B, edge costs are the same as in the original instance, in the first stage as well as the second stage. The costs on the edges between A' and B create the effect of selecting the activated vertices: For each (a'_i, b_i), the first-stage cost is n^2W, and the second-stage cost is n^2W if b is active and 0 otherwise. Here, W is the maximum cost of an edge, nW is an upper bound on the cost of the optimal solution, and n^2W is large enough that any solution containing this edge cannot be an optimal, or even an n-approximate solution. Hence, a second-stage cost of 0 for (a'_i, b_i) allows b_i to be matched with a'_i for free, while a cost of nW forces b_i to be matched with a vertex from A. This concludes the reduction.

Detailed proof of Theorem 2

To define the integer program, let X_e indicate whether edge e is bought in the first stage, and for each scenario s, let Z_e^s (resp. Y_e^s) indicate whether edge e is bought in the first stage (resp. in the second stage) and ends up in the perfect matching when scenario s materializes. We obtain:

$$\min \sum_{s\in S} \Pr(s)(\sum_e C_e X_e + \sum_e C_e^s Y_e^s) \text{ s.t. } \begin{cases} \sum_{e:v\in e}(Z_e^s + Y_e^s) = 1 & \forall v \in A\cup B \text{ and } \forall s \in S \\ Z_e^s \le X_e & \forall e \in E \text{ and } s \in S \\ X_e, Y_e^s, Z_e^s \in \{0,1\} & \forall e \in E \text{ and } s \in S. \end{cases}$$

The algorithm solves the standard linear programming relaxation, in which the last set of constraints is replaced by $0 \le X_e, Y_e^s, Z_e^s \le 1$. Let (X_e, Z_e^s, Y_e^s) denote the optimal solution of the linear program. Now the proof of the two parts of the Theorem diverge: To prove part 1, the algorithm buys, in the first stage, every edge such that $X_e \ge 1/(2n^2)$, and in the second stage, every edge such that $Y_e^s \ge 1/(2n^2)$. To prove part 2, let $\alpha = 8\ln(2)/\beta$. The algorithm buys, in the first stage, every edge e with probability $1 - e^{-X_e\alpha}$, and in the second stage, every edge e with probability $1 - e^{-Y_e^s\alpha}$.

Proof of part 1.

- First stage: the algorithm buys every edge e such that $X_e \ge 1/(2n^2)$.
- Second stage: under scenario s, the algorithm buys every edge e such that $Y_e^s \ge 1/(2n^2)$.

Finally, the algorithm outputs a maximum matching of the set of edges bought.

For the analysis, we see that the expected cost is

$$\sum_{s\in S} \Pr(s)[\sum_{e:X_e\ge 1/(2n^2)} C_e + \sum_{e:Y_e^s\ge 1/(2n^2)} C_e^s] \le 2n^2 \sum_{s\in S} \Pr(s)[\sum_e C_e X_e + \sum_e C_e^s Y_e^s] = 2n^2 \mathrm{OPT}.$$

It only remains to prove that for every scenario s, the output is a perfect matching. By Hall's theorem, since the graph consisting of the edges bought by the algorithm is bipartite, it contains a perfect matching if and only if every subset U of A has at least $|U|$ neighbors in B.

Fix a subset U of A and let $N(U) = \{w \in B | \exists v \in S, P_{\{v,w\}} \ge 1/n^2\}$, where $P_e = Z_e^s + Y_e^s$. Note that if $P_{\{v,w\}} \ge 1/n^2$, then at least one of $Z_{\{v,w\}}^s$ or $Y_{\{v,w\}}^s$ must be greater than or equal to $1/(2n^2)$, and so, the algorithm must have bought edge $\{v,w\}$ under scenario s: thus $N(U)$ is contained in the set of neighbors of U in the graph. Now, since (P_e) is a fractional perfect matching, we have $\sum_{e\in U\times B} P_e = |U|$ and $\sum_{e\in U\times N(U)} P_e \le |N(U)|$. By definition of $N(U)$, we have $\sum_{e\in U\times(B\setminus N(U))} P_e \le |U|(n - |N(U)|)(1/n^2)$, and so

$$|U| \le |N(U)| + \frac{|U|(n-|N(U)|)}{n^2} < |N(U)| + 1.$$

Since $|N(U)|$ is an integer, it must therefore be greater than or equal to $|U|$.

Hence by Hall's theorem there is a matching of size n.

Proof of part 2.

- First stage: the algorithm buys each edge e with probability $1 - e^{-X_e\alpha}$.
- Second stage under scenario s: the algorithm buys each edge e with probability $1 - e^{-Y_e^s\alpha}$.

Finally, the algorithm outputs a maximum matching of the set of edges bought.

For the analysis, we see that the expected cost of the output is

$$\sum_e \left(C_e(1 - e^{-X_e\alpha}) + \sum_s \Pr(s) C_e^s (1 - e^{-Y_e^s \alpha}) \right).$$

Using the upper bound $1 - e^{-Z} \leq Z$, we deduce that this quantity is at most α times the objective function of our linear program, i.e. at most α times OPT.

Let $\beta' = \beta/2$ where we recall that the goal of the theorem is to have a matching of expected size $n - \beta n$. We will prove that with high probability, the output has cardinality at least $n(1-\beta')$. Indeed, assume that the output has cardinality less than $n(1-\beta')$. By König's theorem, since the graph is bipartite, the cardinality of a maximum matching equals the cardinality of a minimum vertex cover [17]. Thus, there exists a set of vertices, of cardinality less than $n(1-\beta')$, which covers all of $E_1 \cup E_2^s$.

Fix a subset V of $A \cup B$ of cardinality less than $n(1-\beta')$. For any edge e that remains uncovered by V, the probability that the algorithm does not buy e is $e^{-X_e\alpha} e^{-Y_e^s \alpha} \leq e^{-P_e \alpha}$, where $P_e = Z_e^s + Y_e^s$. Thus the probability that V is a vertex cover is bounded by $\prod_{e: e \cap V = \emptyset} e^{-P_e\alpha} = e^{-\sum_{e: e\cap V = \emptyset} P_e \alpha}$. By the linear programming constraints and the fact that G is bipartite, $(P_e)_e$ is a convex combination of perfect matchings, each of which has at most $|V|$ edges adjacent to V, hence has at least $\beta' n$ edges not covered by V. Thus the sum of P_e, over edges e left uncovered by V, is at least $\beta' n$. So, the probability that V is a vertex cover is bounded by $e^{-\alpha\beta' n}$.

By the union bound, the probability that there exists such a vertex cover is at most $2^{2n} e^{-\alpha\beta' n} = e^{-(\alpha\beta' - 2\ln 2)n}$. Thus the output matching has size $n(1-\beta')$ with probability at least $(1 - e^{-(\alpha\beta' - 2\ln 2)n})$, and the expected size is at least $(1 - e^{-(\alpha\beta' - 2\ln 2)n})(n(1-\beta')) \geq n(1-\beta)$.

End of proof of Theorem 3

Lemma 4 (Chernoff)**.** *Let $X = \sum_{1 \leq i \leq N} X_i$ be a sum of independent binary random variables, with $E(X_i)$ for all i, and let σ^2 denote the variance of X. Then $\Pr(X - E(X) \geq k\sigma) \leq e^{-k^2/4}$ for any $k \in [0, 2\sigma]$.*

To complete the proof of the Theorem, we apply the Chernoff bound separately for each group with $k = 2\sqrt{3\ln n + \ln t}$, where t is the number of scenarios. For a given group G_ℓ, we have $\sigma^2 = \sum_{i \in G_\ell} p_i(1-p_i) \leq \sum p_i \leq T$; thus, the event that $\sum_{i \in G_\ell} X_i$ is less than $2\sqrt{T(3\ln n + \ln t)}$ has probability at least $1 - 1/(n^3 t)$. The total number of groups is at most $2m \leq n^2$, so with probability at least $1 - 1/(nt)$, the event holds for every group; then we have

$$X \leq \sum_{\ell \geq 1} (\min_{G_\ell} a_i) 2\sqrt{T(3\ln n + \ln t)} \leq 2\alpha \mathrm{OPT}^* (1 + \frac{\ln t}{3\ln n}) \frac{T}{T-1} \leq 3\alpha \mathrm{OPT}^* (1 + \frac{\ln t}{3\ln n}).$$

In other words, with probability at least $1 - 1/(nt)$ the cost is bounded by $3\alpha\mathrm{OPT}^*(1 + \frac{\ln t}{3\ln n})$. Now, recall that t is the total number of scenarios. By the union bound, it holds that with probability at least $1 - 1/n$, we have that, for every scenario, the cost of the algorithm is bounded by $3\alpha\mathrm{OPT}^*(1 + \frac{\ln t}{3\ln n})$. But recall that, by assumption in this section, the number of scenarios is polynomial in n: then $\ln(t) = O(\ln n)$, and the part of the theorem about the cost follows.

For the analysis of the size, it is easy to extend the proof of Theorem 2 so as to show that the output matching has size at least $n(1-\beta/2)$ with probability at least $1 - e^{-(\alpha\beta/2 - \ln 4)n + 2\alpha T}$. Since

$\alpha = o(n/\ln n)$, we have $2\alpha T = o(n \ln 4)$. From our lower bound on $\alpha\beta$, it follows easily that this probability is at least $1 - 1/(nt)$, where t is the number of scenarios. Using the union bound over all scenarios proves that with probability at least $1 - 1/n$, we have that for every scenario the size of the matching satisfies the stated bound of the theorem.

Proof of Theorem 4 for Uncertain Edge Costs

Proof of Part 1. We will prove that when $\tau \geq n^2$, Expression (3) is at least as hard to approximate as Minimum Set Cover: Given a universe $S = \{s_1, \ldots, s_n\}$ of elements and a collection $C = \{c_1, \ldots, c_k\}$ of subsets of S, find a minimum-cardinality subset $\mathcal{SC}$ of C such that for every $1 \leq i \leq n$, $s_i \in c_j$ for some $c_j \in \mathcal{SC}$. It is known that there exists a constant $c > 0$ such that approximating Minimum Set-Cover to within a factor of $c \ln n$ is NP-hard [20].

Given an instance $(S = \{s_1, \ldots, s_n\}; C = \{c_1, \ldots, c_k\})$ of Minimum Set-Cover, we construct an instance of the two-stage matching problem with stochastic matching vertices as follows. The graph contains $|S| + 3|C|$ vertices: for every element $s_i \in S$ there is a vertex u_i; for every set $c_j \in C$, there are three vertices x_j, y_j, and z_j connected by a path $(x_j, y_j), (y_j, z_j)$. For every set c_j and element s_i which belongs to c_j, we have the edge (z_j, u_i). It is easy to see that the graph is bipartite. The first-stage edge costs are 1 for an (x_i, y_i) edge costs and 0 for the other edges. The second-stage costs are equal to the first-stage costs, multiplied by τ. There are n equally likely second-stage scenarios: In scenario i the vertices in $\{y_1, \ldots, y_k\} \cup \{u_i\}$ are active.

Consider a set cover $\mathcal{SC}$ for the input instance. Assume that in the first stage, we buy (at cost 1) the edge (x_j, y_j) for each set $c_j \in \mathcal{SC}$. In the second stage, let i be the scenario and let c_j be a set in the set cover that contains the element s_i. Buy (at cost 0) the edge (z_j, u_i) and every edge $(y_{j'}, z_{j'})$ for $j' \neq j$. Together with the edge (x_j, y_j) which we bought in the first stage, we have a matching that matches all active vertices. The second stage is free in every scenario, so the total cost is equal to $|\mathcal{SC}|$.

On the other hand, assume that the edges bought in the first stage do not correspond to a set cover of the input instance. Let i be an element which is not covered. Then in scenario i, the algorithm will have to match u_i with some z_j such that $u_i \in c_j$, and then it will have to buy the edge (x_j, y_j) at cost n^2. Thus the expected cost is at least $n^2/n = n$.

We get that the minimum of Expression (1) is exactly equal to the cardinality of the minimum set cover of the input instance.

Proof of Part 2. By reduction from the NP-hard Simultaneous Matchings problem [9]: We are given a bipartite graph $G = (X, Y, E)$ and two constraint sets $Z_1, Z_2 \subseteq 2^X$, such that G has a Z_i-perfect matching for each $i = 1, 2$. We need to find a minimum cardinality edge-set $M \subseteq E$ such that for each $i = 1, 2$, $M \cap (Z_i \times Y)$ is a Z_i-perfect matching.

Given an instance of the Simultaneous Matchings problem, we create an instance of our problem as follows. The graph is $G' = (Y, X, E)$. Each edge has cost 1 in the first stage τ in the second stage. There are two equally-likely second-stage scenarios: In scenario i, the vertices of Z_i are active. We show that the instance we created has a solution of cost $\leq |X|$ if and only if the Simultaneous Matching instance has a solution of cardinality $|X|$.

For the first direction, assume that the Simultaneous Matching instance has a solution M of cardinality $|X|$, and consider an algorithm that buys the edges of M in the first stage. The bought edges contain a matching for each second-stage scenario, so the total cost is equal to the first-stage cost, i.e., $|X|$. Conversely, assume that the maximum simultaneous matching in the graph has cardinality smaller than $|X|$ and assume that an algorithm bought the edge-set $\mathcal{M}1$ in the first

stage. If $|\mathcal{M}1| = |X|$, then in at least one of the second-stage scenarios, the matching we can create with the edges of $\mathcal{M}1$ does not match all of the active vertices, so with probability 1/2 we will have to buy at least one edge in the second stage, at cost τ. Finally, if $|\mathcal{M}1| < |X|$ then there are $|X| - |\mathcal{M}1|$ vertices in X that are not incident on any edge in $\mathcal{M}1$. In the second stage, each of these vertices will be active with probability at least 1/2, and in this case we will buy an edge matching it at cost τ. The total cost, then, is at least $|\mathcal{M}1| + (|X| - |\mathcal{M}1|)\tau/2 > |X|$.

A Simpler Proof of Theorem 4, Part 1. for Uncertain Activated Vertices

Recall that Minimum Set Cover, given a collection $C = \{c_1, \ldots, c_k\}$ of subsets of $S = \{s_1, \ldots, s_n\}$, must find the smallest number of subsets from C whose union is S.

Given an instance of set cover, we create an instance of our problem as follows. For every element $s_i \in S$, the graph contains a node u_i. For every set $c_j \in C$, it contains three vertices x_j, y_j, and z_j and the path $(x_j, y_j), (y_j, z_j)$. Additionally, for every set c_j and element s_i such that $s_i \in c_j$, we have the edge (z_j, u_i).

The edge-costs are as follows. In the first stage, the (x_j, y_j) edges have cost 1 and all other edges have cost 0. In the second stage, each edge cost is multiplied by $\tau = n^2$. The active vertices in the second stage belong to one of n scenarios, each of which is realized with probability $1/n$. In Scenario i, the active vertices are $\{y_1, \ldots, y_k\} \cup \{u_i\}$.

Let $\mathcal{SC}$ be a set cover of the input instance. Then there is a solution of value $|\mathcal{SC}|$ to the matching problem we have generated: In the first stage, buy the edge (x_j, y_j) for each $c_j \in SC$. In the second stage, if scenario i is realized, let c_j be a set in $\mathcal{SC}$ that contains u_i; match u_i with z_j and y_j with x_j. Complete into a perfect matching of B_s by matching $y_{j'}$ with $z_{j'}$ for every $j' \neq j$. The second stage is free, so the total cost is $|\mathcal{SC}|$.

On the other hand, if the set of edges (x_j, y_j) bought in the first stage do not correspond to a cover, then let i be an element left uncovered. With probability $1/n$, scenario i occurs in the second stage and the algorithm has to spend at least n^2, hence the expected cost is at least $(1/n)n^2 = n$.

Thus the minimum of Expression (3) is exactly equal to the size of the minimum set cover.

Proof of Theorem 6

As in the proof of Theorem 5, we use a reduction from 3-set-cover(2), the APX-complete special case of set cover where each set has cardinality 3 and each element belongs to two sets. Given an instance of 3-set-cover(2), we create an instance of our problem as follows. For every element $s_i \in S$, the graph contains two vertices u_i and u_i' joined by an edge (u_i, u_i'). For every set $c_j \in C$, it contains three vertices x_j, y_j, and z_j and the path $(x_j, y_j), (y_j, z_j)$. Additionally, for every set c_j and element s_i such that $s_i \in c_j$, we have the edge (z_j, u_i).

The edge-costs are as follows. In the first stage, the (x_j, y_j) edges (x_j, y_j) and (u_i, u_i') have cost 1 and all other edges have cost 0. In the second stage, each edge cost is multiplied by τ. Each vertex in $\{y_1, \ldots, y_k\}$ is activated tomorrow with probability 1 and each vertex in $\{u_1, \ldots, u_n\}$ is activated with probability $p = 1/\tau$. The parameter τ is a constant whose values will be determined later.

Let $\mathcal{SC}$ be a minimum set cover. We construct a solution to the matching problem as follows. In the first stage we buy (at cost 1) the edge (x_j, y_j) for every set $c_j \in \mathcal{SC}$. In the second stage, let I be the set of active vertices, and find, in a way to be described shortly, a matching between a subset I' of elements of I and the sets J' of the set cover $\mathcal{SC}$. Buy (at cost τ) every edge (u_i, u_i') for

$i \in I - I'$, and (at cost 0) every edge (y_j, z_j) for $j \notin J'$. For each matching edge between an element $i \in I'$ and a set $j \in J'$, buy (at cost 0) the edges (u_i, z_j) and complete into a perfect matching of the active vertices by using the first stage edges (x_j, y_j) for $j \in J'$. The second stage has cost equal to τ times the cardinality of $I - I'$ and the first stage has cost equal to the size of the set cover.

The matching is done is a straightforward manner: Given $\mathcal{SC}$, each element chooses exactly one set among the sets covering it, and, if it turns out to be active, will only try to be matched to that set. A set in the set cover, among the active vertices who try to be matched to it, will choose 1 arbitrarily. This defines the matching.

Consider a set $c \in \mathcal{SC}$. We will pay τ for an element in c only if two of its element-vertices are active, and we will pay 2τ only if all three are active. So the expected cost of the second stage is at most $\tau|\mathcal{SC}|(3(1/\tau)^2 + 2(1/\tau)^3)$, and in total the solution costs at most $|\mathcal{SC}|(1 + (3/\tau + 2/\tau^2))$.

On the other hand, for any algorithm, let $\mathcal{M}_1$ be the collection of (x_j, y_j) edges bought in the first stage. If $\mathcal{M}_1$ does not correspond to a set cover, then at least $x \geq |\mathcal{SC}| - \mathcal{M}_1$ elements are uncovered, of which X will be activated, for a minimum second stage cost of $X\tau$ (each must either buy (u_i, u_i') at cost τ or buy some (u_i, z_j) and thus force y_j to buy (y_j, x_j) at cost τ). The cost is at least $\mathcal{M}_1 + \tau X$ which in expectation is $\mathcal{M}_1 + px\tau = \mathcal{M}_1 + x \geq |\mathcal{SC}|$.

In summary, we get that Expression (3) is

$$|\mathcal{SC}| \leq \text{OPT} \leq |\mathcal{SC}|(1 + (3/\tau + 2/\tau^2))$$

and this means that if we can approximate our problem within a factor of β, then we can approximate minimum 3-set-cover(2) within a factor of $\gamma = \beta(1 + (3/\tau + 2/\tau^2))$. It is NP-hard to approximate minimum 3-set-cover(2) to within a factor of 100/99 [4], and with

$$(1/\tau)(3 + 2/\tau) > \frac{1}{99},$$

we get APX-hardness. Note that the inequality holds for $\tau > 1/.0033$.

Proof of Lemma 2

If $\tau p \geq e$, then a vertex in A is miserable with probability $(1-p)^d = (1-p)e^{-\ln(\tau p)} = (1-p)/(\tau p)$. Hence, the expected number of miserable vertices is n times that quantity. Similarly, a vertex in A is poor with probability $dp(1-p)^{d-1} = dp/(\tau p) = d/\tau$. If $\tau p < e$, then a vertex in A is miserable with probability $(1-p)/e$. The lemma follows.

Proof of Lemma 3

Because of the disjoint star structure, the cardinality Z_2 of the maximum-cardinality matching in G_2 is certainly at least $n - E(|A_m|)$ in expectation. Let F be the set of edges bought by OPT in the first stage and B_F denote the set of endpoints of those edges on the B side. The number of edges of F which can be used in the maximum matching is certainly at most $\sum_{b \in B_F} \chi(b \text{ active})$, and so, the cost paid by OPT in the second stage is at least $\tau(Z_2 - \sum_{b \in B_F} \chi(b \text{ active}))$. Thus:

$$\text{OPT} \geq \min\{|F| + \tau(n - E(|A_m|)) - \tau p|F|, |F|\}.$$

If $\tau p \leq 1$ then this expression is minimized for $|F| = 0$, when its value is $\text{OPT} \geq \tau(n - E(|A_m|))$. Otherwise, the expression is minimized for $|F| = (n - E(|A_m|))/p$.

End of proof of Theorem 7

Assume that $\tau p > e$. From Lemmas 1 and 2, it follows that the algorithm has average cost $n(d + 2(1-p)/p + d)$. From Lemmas 2 and 3, it follows that the optimum cost is at least $n(1 - 1/(\tau p))/p$. It follows that the approximation ratio is bounded by

$$\frac{p(d + 2(1-p)/p + d)}{(1 - 1/(\tau p))} = O(1 + dp) = O(\ln(\tau p)\frac{p}{\ln(1/(1-p))}) = O(\ln(\tau p)).$$

On the other hand, assume that $\tau p \leq e$. Then the algorithm has cost at most $n\tau$ and OPT has cost at least $(n - E(|A_m|))\tau/2$. Since $E|A_m| = n/e$, this is $\Omega(n\tau)$ and so the approximation ratio is $O(1)$.