

Approximation Algorithms for Stochastic Combinatorial Optimization Problems

Jian Li* Yu Liu*

December 23, 2015

Abstract

Stochastic optimization has established itself as a major method to handle uncertainty in various optimization problems, by modeling the uncertainty by a probability distribution over possible realizations. Traditionally, the main focus in stochastic optimization has been various stochastic mathematical programming (such as linear programming, convex programming). In recent years, there has been a surge of interest in stochastic combinatorial optimization problems from the theoretical computer science community. In this article, we survey some of the recent results on various stochastic versions of classical combinatorial optimization problems. Since most problems in this domain are NP-hard (or #P-hard, or even PSPACE-hard), we focus on the results which provide polynomial time approximation algorithms, with provable approximation guarantees. Our discussions are centered around a few representative problems, such as stochastic knapsack, stochastic matching, multi-armed bandit etc. We use these examples to introduce several popular stochastic models, such as the fixed set model, 2-stage stochastic optimization model, stochastic adaptive probing model etc, as well as some useful techniques for designing approximation algorithms for stochastic combinatorial optimization problems, including the linear programming relaxation approach, boosted sampling, content resolution schemes, Poisson approximation etc. We also provide some open research questions along the way. Our purpose is to provide the readers a quick glimpse to the models, problems and techniques in this area, and hopefully inspire new contributions.

*Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China 100084. Email: lapordge@gmail.com.

1 Introduction

Uncertainty is an inevitable facet of almost all important decision problems. It is present in a variety of forms and caused by many reasons, such as the errors and/or noises in data measurements, the model parameters, or the predictions made by probabilistic predictive algorithms. Various types of problems of handling uncertainty have been a subject of extensive research in many areas, including computer science, operation research, economics, management science and social science. Stochastic optimization is a major method for dealing with uncertainty for optimization problems, by modeling the uncertainty using probability distributions of the input instances. The field originated from the work of Dantzig [38], and has been studied extensively and found numerous applications in many areas. A central theme in the field is to solve stochastic versions of mathematical programming problems (the study of such problems is often referred to as *stochastic programming* as well). We refer interested readers to the books [22, 124].

Recent years have witnessed a surge of interest in solving combinatorial optimization problems under uncertainty from the theoretical computer science community. This is in part due to the dramatic increase in the number of application domains, such as information extraction systems, data integration systems, sensor networks, and predictive machine learning algorithms, that produces a huge amount of uncertain data. At the same time, applications (including the above) that require solving certain combinatorial optimization problems, when fed with such data, have to take the uncertainty into account. This gives rise to a variety of stochastic models and optimization problems. In this article, we review some recent work on various stochastic versions of classical combinatorial optimization problems. It is well known that many combinatorial optimization problems are NP-hard. So we can imagine that the added stochasticity often makes the problems even harder, (NP-hard, #P-hard, or even PSPACE-hard). Therefore, it is very unlikely that these stochastic problems admit efficient algorithms that can solve them exactly. One principled way to deal with the computational intractability is to settle for polynomial time approximation algorithms. We refer interested readers to the books on this field of approximation algorithms [131, 133]. A major focus in this field is to obtain polynomial time algorithms with provable *approximation ratio*. For maximization problems, the ratio is defined to be the ratio between the optimal cost and the cost of our solution (or expected cost, if any stochasticity or/and randomization¹). For minimization problems, it is the ratio between the cost of our solution and the optimal cost. Obviously, the ratio is at least 1 and we want it to be as close to 1 as possible.

In this article, we present a selection of recent results in polynomial time approximation algorithms for stochastic combinatorial optimization problems. We have not attempted to be comprehensive and our selection of problems may be idiosyncratic. We also include the details of the algorithms and the analysis of approximation ratios for some problems. None of these results are really new. The expositions for some of them are simplifications of known results (possibly with slightly worse approximation ratios sometimes in favor of expositional simplicity). The purpose of doing this is to give the readers a flavor of the models, ideas and techniques in this new exciting and fast growing area.

¹We usually use “stochastic/probabilistic” to indicate that the input instance follows certain probability distribution, while “randomization” to refer to the randomness produced by the randomized algorithms.

1.1 A Few Representative Problems

There are diverse ways in which uncertainty can interact with and influence the decision process. So, there are several natural corresponding stochastic models. Instead of introducing these abstract models, we choose to be concrete, by providing a few representative problems. We also highlight some interesting motivating applications for those problems.

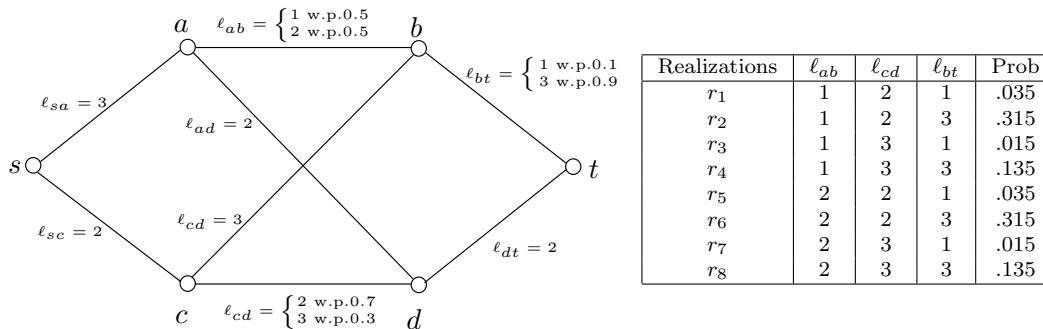


Figure 1: A stochastic graph. There are three edges with uncertain lengths. There are $2^3 = 8$ possible realizations in total.

Problem 1.1. (*Fixed Set Stochastic Shortest Path*): We are given a stochastic graph where the length μ_e of each edge e is an independent random variable.² See Figure 1 for an example. The objective is to find an s - t path P connecting s and t such that the probability that the length of P is at most a given threshold \mathbb{C} , i.e., $\Pr[\sum_{e \in P} \mu_e \leq \mathbb{C}]$, is maximized.³

We can also think μ_e as the travel time of the edge e (the probability distribution of μ_e may be obtained from historic data, or traffic prediction algorithms). We need to arrive the airport before a specific time, in order to catch a flight. The above problem asks for a path that maximizes the probability that we can arrive on time.

The problem and its variants have been studied in a number of papers since 1980s (see e.g., [109, 14, 119, 118, 106, 108]). Besides the shortest path problem, one can easily formulate similar problems for other combinatorial optimization problems. We will discuss this problem and its generalizations in more details in Section 8.

Note that the solution path P is chosen before the trip, and once chosen, is a fixed set of edges. We call such problems *fixed set* problems. Next, we introduce a problem in which the decision has to be made in an adaptive manner.

Problem 1.2. (*Adaptive Stochastic Knapsack*) We are given a knapsack with fixed capacity \mathbb{C} and a set of items. The size and the profit of each item are uncertain and only their probability distributions are known to us in advance. We only know the actual size and profit of an item when we insert it into the knapsack. Our goal is to design an adaptive policy that maximizes the expected

²For simplicity, we assume the distribution of each random variable is discrete with a finite support. The complete description of the distribution is given as input.

³Another natural problem would be to minimize the expected length of the path. However, a moment reflection shows that the problem can be trivially reduced to deterministic shortest path problem, using the linearity of expectation.

total profit of items that can be put into the knapsack. More specifically, the adaptive policy should determine which item to insert next based on the remaining capacity and the set of available items.

The above problem has a very natural motivating example in stochastic scheduling. Suppose we want to schedule a subset of n jobs on a single machine. We have a fixed deadline and we want to gain as much profit as possible before the deadline. The precise processing time and profit of each job are random variables, and the true values are only revealed until the job is completed.

The problem is a classical problem in stochastic combinatorial optimization. It was studied back in 70s in the stochastic control and operation research literature, and some interesting special cases were identified and solved optimally (see e.g., [42, 43, 9, 8]). The problem has a very different nature from the fixed set problems, in that a solution to it is an *adaptive policy*, instead of a set. Even to specify the optimal policy may need exponential space (see Section 3 for details). The problem is likely to be PSPACE-hard (a variant of the problem has been shown to be PSPACE-hard [16]). Designing good approximation algorithms for this problem and its generalizations has received considerable attention in recent years, which we review in more details in Section 3.

In the fixed set stochastic shortest path problem, we make our final decision in one shot, while in the stochastic knapsack problem, we make decisions in an adaptive fashion. Another popular model that interpolates the above two models is the two-stage stochastic model with recourse. We provide an example below.

Problem 1.3. (*Two-stage Stochastic Set Cover*) We are given a set U of ground elements and the weighted family \mathcal{F} of subsets of U . We would like to choose some subsets $\mathcal{S} \subseteq \mathcal{F}$ of minimum total weight to cover all elements in U (i.e., each element of U is contained in some subset of \mathcal{S}). This is the classical deterministic set cover problem. In the two-stage stochastic model, there are two stages (by its name). In the first stage, we do not know the actual demand (i.e., the set of elements we need to cover), but only a distribution \mathcal{D} of it.⁴ We can select a few subsets $\mathcal{S}^I \subseteq \mathcal{F}$ at the first stage. In the second stage, the actual demand set U' ($U' \subseteq U$) is revealed. However, our first stage decision \mathcal{S}^I may not be able to cover U' , and we can select some extra subsets $\mathcal{S}^{II} \subseteq \mathcal{F}$ to cover the remaining elements of U' . The catch is that the weight of each subset in \mathcal{F} in the second stage is λ times more expensive than in the first stage ($\lambda > 1$ is called the inflation factor).⁵ The goal is to minimize the expected cost over both stages.

The above 2-stage model naturally captures the applications in which the decision has to be made before knowing the demand, and some more expensive recourse action can be taken afterwards. For example, suppose there is a company which needs to set up its facilities. Before building the facilities, they only know the partial knowledge about the demands. After setting up the facilities, they receive the real demand. If the demand exceeds the capability of the current facilities, the company can set up more facilities, which is usually more expensive, to meet the extra demand.

The two-stage stochastic optimization model has been a major topic in both the stochastic programming literature (see e.g., [22]), and in the stochastic combinatorial optimization literature, which we discuss in more details in Section 6 and 7.

⁴We may assume that each element $e \in U$ is associated with a probability $p_e \in [0, 1]$, which means e is present in the demand set with probability p_e independently. The current technique can handle more general distributions. See Section 6.

⁵It is easy to see that if $\lambda \leq 1$, the first stage is meaningless and we only need to make the decision in the second stage.

All the above three problems assume the stochastic information is given as the input. However, in many real life problems, there is no readily available stochastic information. All we have are some presumed probabilistic ⁶ models with unknown parameters (e.g., the cost is distributed normally, with unknown mean and variance). We are allowed to take samples (or we may already have some historical samples) to obtain the distributional information (e.g., to learn the parameters of the probabilistic model). The learning step has traditionally concerned the statistical learning community, and was separated from the optimization stage in many cases. However, there have been several interesting recent research problems that attempt to bring the learning step and the optimization step together. We provide an example below. More can be found in Section 9.

Problem 1.4. (*Combinatorial Bandit Arm Selection*) *We have a bandit with n arms, where the i -th arm is associated with an unknown reward distribution supported on $[0, 1]$ with mean θ_i . Upon each sample (or “pull”) of a particular arm, we receive a reward, which is an i.i.d. sample from the underlying reward distribution. We sequentially decide which arm to be pulled next and then collect the reward by sampling that arm. The top- k arm identification problem is to identify a subset of k arms with the maximum total mean, using as few samples as possible.*

Instead of the simple cardinality constraints, we can consider other combinatorial constraints over the arms as well. For example, suppose there is a matroid constraint over the set of arms and we need to select a basis of the matroid. The model was proposed recently by Chen et al. [36].

We highlight an interesting motivating application of the top- k arm identification in *crowdsourcing*. In recent years, crowdsourcing services become increasingly popular for providing labeled data for many machine learning tasks. In such a service (e.g., Amazon Mechanical Turk ⁷), the requester submits a batch of microtasks (e.g., unlabeled data) and the workers from the crowd are asked to complete the tasks. Upon each task completion, a worker receives a small monetary reward. Since some workers from the crowd are noisy and unreliable, it is important to choose more reliable workers. We can model the reliability of a worker by a Bernoulli distribution. One way to choose reliable workers is to test each worker by a few *golden samples* (i.e., data with the known correct labels). One such test corresponds to a pull of the arm. Clearly, it is desirable to select the best K workers using as few samples as possible.

The reader may have already noticed the deterministic versions of the above problems are mostly classical combinatorial optimization problems, which are quite well understood. However, the stochasticity has drastically changed the nature of these problems. The study of the approximability for many of stochastic combinatorial problems is still an active research area and there are many open problems. We attempt to list some of them along the way.

We note that the body of literature on the topic is already large and has been growing quite fast. There are certainly other important stochastic combinatorial optimization problems that do not belong to any of the above four classes. Depending on how the stochasticity interacts with the decision process in new applications, it is possible to formulate new classes of meaningful models and problems. We mention some of them in Section 9.

Outline: We first review some standard terminologies in Section 2. In the next few sections, we discuss a few problems and techniques in details. In particular, we discuss the stochastic knapsack

⁶We use “probabilistic” and “stochastic” interchangeably, except when there is a convention to follow (e.g., “stochastic process”, “stochastic optimization”, “probabilistic graphical model” etc.).

⁷<https://www.mturk.com>

problem (Problem 1.2) in Section 3, the stochastic matching problem in Section 4, and the contention resolution scheme with its applications to stochastic probing in Section 5. The above three sections are about adaptive stochastic problems. In the next two sections, we discuss two important techniques, the LP technique (Section 6) and the boosted sampling technique (Section 7) for two stage stochastic optimization with recourse. In Section 8, we introduce the Poisson approximation technique, which can be used to deal with the sum of random variables in both fixed set problems and stochastic knapsack. In Section 9, we briefly discuss a few other important models in the literature.

2 Preliminaries

We first review some standard terminologies. For a minimization problem, we say an algorithm achieves an approximation factor of α ($\alpha \geq 1$), if

$$\mathbb{E}[\text{SOL}]/\text{OPT} \leq \alpha,$$

where SOL denotes the cost of the solution found by the algorithm, OPT denotes the optimal cost, and the expectation is over the randomness of the problem instance and the algorithm (if it is randomized). For a maximization problem, an algorithm achieves an approximation factor of α ($\alpha \geq 1$), if $\text{OPT}/\mathbb{E}[\text{SOL}] \leq \alpha$.

A *polynomial time approximation scheme (PTAS)* is an algorithm which takes an instance of a maximization problem and a parameter ϵ and produces a solution whose cost is at least $(1 - \epsilon)\text{OPT}$, and the running time, for any fixed ϵ , is polynomial in the size of the input. If ϵ appears as an additive factor in the above definition, namely the cost of the solution is at least $\text{OPT} - \epsilon$, we say the algorithm is an *additive PTAS*. In many cases, a PTAS (such as an $O(n^{1/\epsilon^2})$ time algorithm) is not efficient in practice. However, obtaining a PTAS for a problem is of significant theoretical interest and importance, as it is the best approximation ratio we can achieve in polynomial time for NP-hard (or harder) problems. We say a PTAS is a *fully polynomial time approximation scheme (FPTAS)* if the running time is polynomial in the size of the input and $\frac{1}{\epsilon}$. FPTAS can be quite efficient in practice.

For a deterministic combinatorial optimization problem \mathbb{A} , the *exact version* of a problem \mathbb{A} asks for a feasible solution of \mathbb{A} with weight exactly equal to a given number K . An algorithm runs in *pseudopolynomial time* for the exact version if the running time of the algorithm is bounded by a polynomial of n and K .

The following standard Markov inequality will be used frequently.

Proposition 2.1. (*Markov inequality*) *Let X be a random variable and $\mathbb{E}[X]$ be its expectation. For any $\alpha > 1$, it holds that*

$$\Pr[X \geq \alpha\mathbb{E}[X]] \leq 1/\alpha.$$

A finite matroid \mathcal{M} is a pair (V, \mathcal{I}) , where V is a finite set (called the *ground set*) and \mathcal{I} is a collection of subsets of V . Each element in \mathcal{I} is called an *independent set*. A maximal independent set is called a *basis*. Moreover, $\mathcal{M} = (V, \mathcal{I})$ satisfies the following three properties:

1. $\emptyset \in \mathcal{I}$;
2. if $A \subseteq B$ and $B \in \mathcal{I}$, then $A \in \mathcal{I}$;

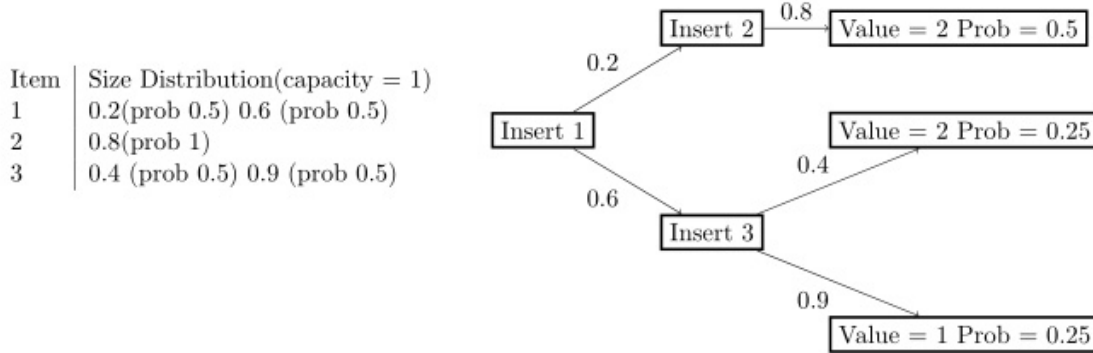


Figure 2: A decision tree corresponding to an adaptive policy.

- for all $A, B \in \mathcal{I}$ with $|A| > |B|$, there exists an element $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$.

Matroids generalize several important combinatorial objects, such as all subsets of a set with cardinality at most k , all forests in a graph, all linear independent subsets of a set of vectors. For more information about the theory of matroids, see, e.g., [123].

3 Adaptive Stochastic Knapsack

In this section, we consider Problem 1.2 the adaptive stochastic knapsack problem. First, we recall in the deterministic knapsack problem, we have a set of n item \mathcal{I} . The i th item has a value v_i and size s_i . Our objective is to choose a maximum-value set of items that fits into a knapsack of capacity \mathbb{C} .

In the stochastic knapsack problem, the values $\{v_i\}$ and sizes $\{s_i\}$ of items are non-negative independent random variables, with known distributions. For simplicity, we assume the distributions are discrete. One needs to insert the items one by one. Once one item is inserted, its value and size are revealed (drawn from the distribution), and we procure its value. If the realization of an item causes the overflow of the knapsack, we stop and the overflowing item contributes zero value. Our goal is to design a policy to insert items and maximize the expected total value of items successfully placed in the knapsack. Without loss of generality, we can assume that the values $\{v_i\}$ are deterministic, and the capacity of the knapsack is $\mathbb{C} = 1$.

We need to explain what is a policy. A *non-adaptive* policy is simply an ordering of items to insert. We also consider *adaptive policies*, which are more general. An adaptive policy is formally defined by a mapping $\mathcal{P} : 2^{\mathcal{I}} \times R^+ \rightarrow \mathcal{I}$ specifying which item to insert next, given the subset of items still available and the total remaining capacity. We can think about a policy as a decision tree. At every node of the tree, the policy specifies which item to insert next, and the directed edges leaving this node correspond to the different possible realizations of the size of this item. See Figure 2 for an example. We can see the decision tree corresponding to an adaptive policy may have a linear height, and thus an exponential size. This means that we may not be able to represent the optimal policy in polynomial space. So, it is not even clear if the problem is in NP. In fact, Dean et al. [40] showed that a close variant of the above problem is PSPACE-hard.

Now, we present a constant factor approximation algorithm, due to [16]. Before stating the algorithm, we first need some definitions and simple observations. For each item, it is convenient

to consider the mean truncated size

$$\mu_i = \mathbb{E}[\min\{s_i, 1\}].$$

For a set of item, we define $size(S) = \sum_{i \in S} s_i$, $\mu(S) = \sum_{i \in S} \mu_i$ and $val(S) = \sum_{i \in S} v_i$. We make the following simple claim.

Claim 3.1. *For any fixed set S of items, $\Pr[size(S) < 1] \geq 1 - \mu(S)$.*

Proof. By the definition of mean truncated size and Markov inequality, we can see that $\Pr[size(S) \geq 1] = \Pr[\min\{size(S), 1\} \geq 1] \leq \mathbb{E}[\min\{size(S), 1\}] \leq \mathbb{E}[\sum_{i \in S} \min\{s_i, 1\}] = \mu(S)$. ■

Lemma 3.2. *For any adaptive policy \mathcal{P} , let S be the (random) set of all items that \mathcal{P} tries to insert. We have $\mathbb{E}[\mu(S)] \leq 2$, where the expectation is taken over the executions of the policy.*

Proof. (sketch) Denote the truncated item size by $\tilde{s}_i = \min\{s_i, 1\}$. For any policy \mathcal{P} , let S_t denote the first t items chosen by \mathcal{P} . We define $X_t = \sum_{i \in S_t} (\tilde{s}_i - \mu_i)$. It is easy to verify X_t is a martingale.⁸ Since $X_0 = 0$, we get $\mathbb{E}[X_t] = \mathbb{E}[X_0] = 0$ due to the martingale property. Thus, $\mathbb{E}[X_t] = \mathbb{E}[\sum_{i \in S_t} \tilde{s}_i] - \mathbb{E}[\mu(S_t)] = 0$. Moreover, the process stops once $size(S_t) > 1$ or there is no item left. Hence, $\sum_{i \in S_t} \tilde{s}_i \leq 2$ is always true. Taking the expectation, we can see $\mathbb{E}[\mu(S_t)] \leq 2$. ■

3.1 The Greedy Algorithm

We divide the items \mathcal{I} into two sets: light items \mathcal{L} with $\mu_i \leq \sigma$ and heavy items \mathcal{H} with $\mu_i > \sigma$, where σ is a constant to be specified later. Assume $\mathcal{L} = \{1, 2, 3, \dots\}$ such that $\frac{v_1}{\mu_1} \geq \frac{v_2}{\mu_2} \geq \dots$. Denote by $M_k = \sum_{i=1}^k \mu_i$. Let n^* be the maximum number such that $M_{n^*} \leq 1$. We define two important quantities

$$m_G = \sum_{i=1}^{n^*} v_i(1 - M_i) \quad \text{and} \quad m_1 = \max_{i \in \mathcal{I}} \{v_i \Pr[s_i \leq 1]\}.$$

The algorithm is the following:

Algorithm 1: Greedy Algorithm

- 1 Calculate m_1, m_G ;
 - 2 **if** $m_1 \geq m_G$ **then**
 - 3 Insert the item which yields $m_1 = v_i \Pr[s_i \leq 1]$;
 - 4 **else**
 - 5 **for** $i = 1 \rightarrow n^*$ **do**
 - 6 Insert item i ;
 - 7 If the knapsack overflows, break;
-

⁸ $\{X_t\}$ is a martingale if $\mathbb{E}[X_{t+1} | X_t] = X_t$.

3.2 Analysis

Now, we analyze the approximation ratio of the greedy algorithm. First, we show the expected value of the algorithm can be lower bounded by m_G and m_1 .

Lemma 3.3. *Let GRD be the expected value obtained by Algorithm 1. It holds that $\text{GRD} \geq m_1$ and $\text{GRD} \geq m_G$.*

Proof. If $m_1 \geq m_G$, the expected value of the algorithm is exactly m_1 . Otherwise, we can see

$$\text{GRD} = \sum_{k \geq 1} v_k \Pr \left[\sum_{i=1}^k s_i \leq 1 \right] = \sum_{k \geq 1} v_k \Pr \left[\sum_{i=1}^k \tilde{s}_i \leq 1 \right].$$

By Markov inequality, $\Pr[\sum_{i=1}^k \tilde{s}_i > 1] \leq \sum_{i=1}^k \mu_i = M_k$, for any $k \leq n^*$. Hence, $\text{GRD} \geq \sum_{k=1}^{n^*} v_k (1 - M_k) = m_G$. ■

The remaining task is to lower bound m_1 and m_G by the optimal value OPT. We handle light items and heavy items separately. We first deal with light items \mathcal{L} . We define a function $\Phi(c)$ as follows:

$$\Phi(c) = \left\{ \max \sum_{i \in \mathcal{L}} v_i x_i : 0 \leq x_i \leq 1, \sum_{i \in \mathcal{L}} \mu_i x_i \leq c \right\}. \quad (1)$$

We can see $\Phi(c)$ is the best fractional relaxation of the knapsack problem with capacity c , using only light items. A crucial observation is the following lemma, which places an upper bound for any adaptive policy.

Lemma 3.4. *Let $\text{OPT}_{\mathcal{L}}$ be the expected value obtained by the optimal policy for the problem where we only have light items. Then, we have $\text{OPT}_{\mathcal{L}} \leq \Phi(2)$.*

Proof. (sketch) Let us fix an arbitrary adaptive policy. We should interpret x_i as the probability that item i is successfully placed in the knapsack by the policy. So, $\sum_{i \in \mathcal{L}} v_i x_i$ is the expected total value obtained by the policy. According to Lemma 3.2, it holds that $\sum_{i \in \mathcal{L}} \mu_i x_i = \mathbb{E}[\mu(S)] \leq 2$, where S is the set of items successfully placed in the knapsack. This constraint holds for any policy, including the optimal one. Hence, the optimal policy has expected value at most $\Phi(2)$. ■

It is well known that the optimal fractional solution $\Phi(c)$ for the knapsack problem: we simply pack items in the decreasing order of $\frac{v_i}{\mu_i}$, with only the last item possibly packed fractionally. In particular, $\Phi(c)$ is a piecewise linear and concave function, with the following explicit form:

$$\forall k \in \mathcal{L}, \forall \delta \in [0, \mu_k), \Phi(M_{k-1} + \delta) = \sum_{i=1}^{k-1} v_i + \frac{v_k}{\mu_k} \delta.$$

Lemma 3.5. $m_G \geq \frac{1-\sigma}{2} \Phi(1) \geq \frac{1-\sigma}{4} \Phi(2) \geq \frac{1-\sigma}{4} \text{OPT}_{\mathcal{L}}$.

Proof. We prove the lemma by picture. m_G corresponds to the area under the staircase. $\Phi(1)$ is twice of the area of the dark shaded triangle ABC. Compare them in the Figure 3.2. The part of the triangle which is not covered by m_G consists of small triangles whose heights sum up to at most $\Phi(1)$, and the base of each triangle is $\mu_i \leq \sigma$. Therefore $m_G \geq \frac{1-\sigma}{2} \Phi(1)$. The second inequality is due to the concavity of Φ and the last follows from Lemma 3.4. ■

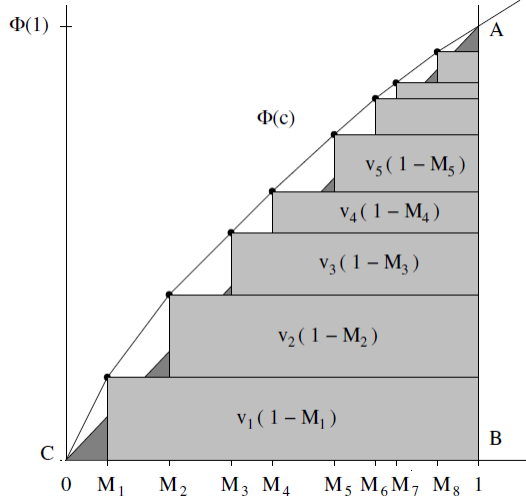


Figure 3: The fractional knapsack solution $\Phi(c)$. The light shaded area corresponds to m_G . As $\Phi(c)$ is concave, the curve $\Phi(c)$ is above the segment AC .

Lemma 3.6. *For any policy \mathcal{P} , we use $H \subset \mathcal{H}$ to denote the subset of heavy items that are successfully placed in the knapsack. Note that H is a random set. Then $\mathbb{E}[\text{val}(H)] \leq \frac{2}{\sigma}m_1$.*

Proof. Whenever the policy \mathcal{P} attempts to insert an element, it succeeds with probability at most $\Pr[s_i \leq 1]$. Let S be the set of heavy item that \mathcal{P} attempts to insert. Thus we have $\Pr[i \in H] \leq \Pr[\mathcal{P} \text{ attempts to insert } i] \Pr[s_i \leq 1] = \Pr[i \in S] \Pr[s_i \leq 1]$. From Lemma 3.2, we know $E[\mu(S)] \leq 2$. Finally, we can see that

$$\begin{aligned} \mathbb{E}[\text{val}(H)] &= \sum_i v_i \Pr[i \in H] \leq \sum_i v_i \Pr[i \in S] \Pr[s_i \leq 1] \\ &\leq m_1 \sum_i \Pr[i \in S] = \mathbb{E}[|S|]m_1 \leq \frac{2}{\sigma}m_1. \end{aligned}$$

The last inequality follows from by the definition of heavy element, $\mu_i \geq \sigma$ for each $i \in S$. ■

Theorem 3.7. *For $\sigma = 1/3$, $\text{OPT} \leq 12\text{GRD}$. So, Algorithm 1 is a 12-approximation.*

Proof. Consider the optimal adaptive policy, and denote the set of items successfully inserted as $S = L \cup H$. Therefore, by Lemma 3.5, $\mathbb{E}[\text{val}(L)] \leq \text{OPT}_{\mathcal{L}} \leq \frac{4}{1-\sigma}m_G$, and by Lemma 3.6, $\mathbb{E}[\text{val}(H)] \leq \frac{3}{\sigma}m_1$. Combining with Lemma 3.3, we obtain that

$$\text{OPT} = \mathbb{E}[\text{val}(L)] + \mathbb{E}[\text{val}(H)] \leq \frac{4}{1-\sigma}m_G + \frac{2}{\sigma}m_1 \leq 12\text{GRD}. \quad \blacksquare$$

3.3 Historical Notes, Variants, and Generalizations

The adaptive stochastic knapsack was first studied by Dean, Goemans and Vondrak [16]. The above greedy algorithm and the analysis are essentially due to them. With some extra effort, one

can in fact show the greedy algorithm achieves an approximation factor of 7. Note that the greedy algorithm is in fact a *non-adaptive* policy, and we compare it against the optimal adaptive policy (which may not have a polynomial size representation). The crux of the analysis, that allows us to upper bound the optimal value, is Lemma 3.4. Although the approach is very similar to the linear programming relaxation technique used to approximate deterministic NP-hard problems, they have essential difference. In the deterministic case, we have the integer constraints on the variables, and we relax them. In our case, the variables should be interpreted as certain probabilities, which can be fractional. But we only have some necessary (but not sufficient) constraints on the probabilities (in the above problem, the constraint is $\sum_{i \in \mathcal{L}} \mu_i x_i \leq 2$). In Section 4, we will see another example of this idea.

In the journal version of their paper [41], they presented a randomized variant of the greedy algorithm which can achieve a factor of $32/7$. In the same paper, they proposed another useful idea that can further improve the approximation factor. Note that in expectation, the number of heavy items that can be inserted successfully is bounded by a small constant. Hence, it is possible to enumerate all possible ways to insert heavy items (by enumerating the decision tree). Using this idea, they managed to improve the approximation factor to 3.

The idea of enumerating the decision trees was carried further by Bhargat, Goel and Khanna [20]. They showed that there exists a special class of policies called *block adaptive policies* which, combined with their discretization technique, allows one to enumerate all possible decision trees for such policies in polynomial time. They used this idea to obtain a *bi-criterion PTAS*: their policy can achieve an expected value of $(1 - \epsilon)\text{OPT}$, and the knapsack capacity may be violated by at most ϵ , for any constant $\epsilon > 0$. Typically, such a result can not be achieved by the LP technique. Using this result, Bhargat [19] obtained a $(2 + \epsilon)$ -approximation, which is currently the best known approximation factor for the adaptive stochastic knapsack problem. Their approach is further simplified and generalized in [108], using the Poisson approximation technique, which we will detail in Section 8.

Stochastic Knapsack with Correlations and Cancellations: Gupta et al. [74] considered a generalization, in which the value and the size of a job can be correlated (described by a joint distribution), and we can cancel a job in the middle. They provided an approximation algorithm with an approximation factor of 8, based on a new time-indexed LP relaxation. They also considered a further generalization, called the non-martingale bandit problem, and provided a constant factor approximation algorithm. Using the Poisson approximation technique, Li and Yuan [108] obtained a bi-criterion PTAS for the stochastic problem with correlations and cancellations. They also show that the bi-criterion PTAS can be easily converted into polynomial time $(2 + \epsilon)$ -approximation if we only allow cancellations. Ma [110] again used the LP technique, but with a different LP which corresponds to a dynamic program, and obtained a polynomial time algorithm that finds a $(2 + \epsilon)$ -approximate adaptive policy for stochastic knapsack with correlations or cancellations, if all the possible sizes of any item are integers. We conclude the discussion of the stochastic knapsack problem with the following obvious open questions:

Open Question 1. *Is there a PTAS for the adaptive stochastic knapsack problem (without violating the capacity constraint)?*

Note that it is known that certain variants of stochastic knapsack are PSPACE-hard to approximate within some constant factors [16, 40]. However, these results do not preclude an affirmative answer of the above question. Obtaining a better-than-2 approximation ratio is also a very inter-

esting open question, which seems to require new ideas. In fact, even for the basic version of the adaptive stochastic knapsack, where the sizes are random but the values are deterministic, whether the problem is PSPACE-hard is still open.

Stochastic Orienteering Problem: Stochastic orienteering is a generalization of stochastic knapsack. It is defined as follows:

Problem 3.8. (*Stochastic Orienteering*) We first introduce the deterministic version. We are given a metric (V, d) , where V is the set of vertices and d is the distance metric, a budget B and an initial vertex $r \in V$. Each vertex has a job associated with a reward and processing time. We assume that the travel time between two vertices u and v is proportional to $d(u, v)$. The goal is to compute a path starting from r that visits a subset of vertices and run the jobs on these vertices, so as to maximize the total reward, subject to the constraint that the travel time plus the job processing time do not exceed the budget B .

In the stochastic version, the reward and the processing time of each job are random variables. The distributions are also given as input. The goal now is to compute an adaptive policy that maximizes the expected total reward.

We can easily see that if all the distances are 0, the problem reduces to the stochastic knapsack problem. The problem is introduced in Gupta et al. [74]. They also showed that when the rewards are deterministic, there is a policy which is an $O(1)$ -approximation to the best non-adaptive policy (that is to visit the vertices in a fixed order), while it is an $O(\log \log B)$ -approximation to the best adaptive policy. It was attempting to believe that there is a non-adaptive policy which is a constant approximation of the best adaptive policy. Somewhat surprisingly, Bansal and Nagarajan [13] showed the gap between the best adaptive policy and the best non-adaptive policy is at least $\Omega((\log \log B)^{1/2})$, if the rewards are deterministic.

4 Stochastic Matching

In this section, we consider the following adaptive stochastic matching problems. We will provide a simple constant factor approximation algorithm using the LP-rounding technique. In Section 5, we will present a significant extension of this technique, called the contention resolution scheme, which can be used to handle several other combinatorial constraints as well.

4.1 Problem Definition

Problem 4.1. (*Adaptive Stochastic Matching*) We are given a probabilistic graph $G(V, E)$ on n nodes. We would like to find a maximum weighted matching in this graph. For each edge (u, v) , we are given a probability value p_{uv} , a weight w_{uv} . Each vertex u is associated with a positive integer t_u (the patience level). To find out whether the edge (u, v) is present in G or not, we can “probe” the pair (u, v) (assuming that u and v are still unmatched). If the edge (u, v) indeed exists (which happens with probability p_{uv} , independent of other edges), it gets irrevocably added to our matching M . Moreover, we can probe at most t_u edges incident to vertex u (t_u can be thought as the patience level of vertex u). Our goal is to design an adaptive probing policy that maximizes the expected total weight of the matching M .

The problem is motivated by the kidney exchange program, which we now briefly describe. The problem and some of its variants also find important applications in the popular online dating application and online advertisement assignment (see [12] for more details).

A motivating example: Kidney Exchange: *The United Network for Organ Sharing (UNOS) launched in year 2000 the famous kidney exchange program, which has saved thousands of lives [122].⁹ We briefly describe the program and how it can be modeled as the stochastic matching problem. Suppose a family member of the patient would like to donate a kidney to the patient but the kidney is incompatible with the patient’s body. The idea is to identify two incompatible patient/donor pairs such that each donor is compatible with the other pair’s patient, and then perform the kidney exchange between the two pairs. We can think each patient/donor pair as a vertex, and there is an edge between two vertices if the compatibility allows for an exchange. Of course, our goal is to find a large matching. To decide compatibility, three main tests must be performed. The first two tests, the blood-type test and the antibody screen, are fairly easy to perform, while the third, called crossmatching, is the most critical one, yet very expensive and time-consuming. However, the compatibility can only be determined by this test. Therefore, if the crossmatch test is passed for a pair, the transplant should be performed immediately. Thus, we can estimate the probability that two pairs are compatible based on the initial two tests, model it by a probabilistic edge. A crossmatch test corresponds to a probe on an edge. If the probe is successful, we should include this edge in our matching (i.e., the exchange should be performed). The patience level for each vertex is also very natural: it models the fact that a patient will eventually die without a successful match.*

4.2 Algorithm

Similar as the stochastic knapsack problem, we use a linear program to upper bound the optimal profit, and to guide our algorithm. Consider the following LP: for any vertex $v \in V$, $\partial(v)$ denotes the edges incident to v . Variable y_e denotes the probability that edge $e = (u, v)$ gets probed in the adaptive strategy, and $x_e = p_e \cdot y_e$ denotes the probability that u and v get matched in the strategy.

$$\text{maximize } \sum_{e \in E} w_e \cdot x_e \tag{2}$$

$$\text{subject to } \sum_{e \in \partial(v)} x_e \leq 1 \quad \forall v \in V \tag{3}$$

$$\sum_{e \in \partial(v)} y_e \leq t_v \quad \forall v \in V \tag{4}$$

$$x_e = p_e \cdot y_e \quad \forall e \in E \tag{5}$$

$$0 \leq y_e \leq 1 \quad \forall e \in E \tag{6}$$

Claim 4.2. *The optimal solution of the above LP is an upper bound of the optimal expected profit of any adaptive policy, which we denote as OPT.*

We first solve the LP to the optimal and let $\{y_e^*\}_{e \in E}$ denote an optimal solution to this linear program. Let $x_e^* = p_e \cdot y_e^*$.

⁹The founder of the program, Alvin E. Roth, won the Nobel Prize in Economics in 2012, partly due to the implementation of the program.

Our rounding algorithm proceeds as follows. Fix a constant $\alpha \geq 1$, to be specified later. The algorithm picks a uniformly random permutation $\pi : E \rightarrow E$ on all edges. Then, it inspects the edges in the order of π . We say an edge $e = (u, v)$ is *safe* if both u and v are unmatched and both t_u and t_v are larger than zero. Whenever it is *safe* to probe the next edge $e \in E$, the algorithm does so with probability $\frac{y_e^*}{\alpha}$. If edge e indeed exists, we include e in the matching. Otherwise, we decrease the patience levels of both endpoints of e by 1. Note that the algorithm skips all unsafe edges at the time they appear in π . See Algorithm 2 for the pseudocode.

Algorithm 2: Stochastic Matching

- 1 Pick a permutation π on edges E uniformly at random;
 - 2 **for** *For each edge e in the ordering π do*
 - 3 If $e = (u, v)$ is not safe then do not probe it;
 - 4 If e is safe then probe it with probability y_e/α , else do not probe it;
-

4.3 Analysis

The approximation guarantee is summarized in the following theorem.

Theorem 4.3. *For $\alpha = 4$, Algorithm 2 achieves an approximation factor of 8 against any adaptive policy for the stochastic matching problem.*

We start with a simple lemma, which bounds the probability of the safeness of an edge when it is considered by the algorithm.

Lemma 4.4. *For any edge $(u, v) \in E$, at the point when (u, v) is considered under π ,*

- (a) *the probability that vertex u loses its patience is at most $\frac{1}{2\alpha}$, and*
- (b) *the probability that vertex u is matched is at most $\frac{1}{2\alpha}$.*

Proof. We begin with the proof of part (a). Let random variable U denote the number of probes incident to vertex u by the time edge (u, v) is considered in π .

$$\begin{aligned} \mathbb{E}[U] &= \sum_{e \in \partial(u)} \Pr[\text{edge } e \text{ appears before } (u, v) \text{ in } \pi \text{ and } e \text{ is probed}] \\ &\leq \sum_{e \in \partial(u)} \Pr[\text{edge } e \text{ appears before } (u, v) \text{ in } \pi] \cdot \frac{y_e^*}{\alpha} = \sum_{e \in \partial(u)} \frac{y_e^*}{2\alpha} \leq \frac{t_u}{2\alpha}. \end{aligned}$$

The first inequality above follows from the fact that the probability that edge e is probed (conditioned on π) is at most y_e^*/α . Since π is a uniformly chosen random permutation on E , edge e appears before (u, v) with probability $1/2$. Hence the second equality follows. The last inequality is by the LP constraint (4).

By Markov inequality, the probability that vertex u has timed out when (u, v) is considered equals

$$\Pr[U \geq t_u] \leq \frac{\mathbb{E}[U]}{t_u} \leq \frac{1}{2\alpha}.$$

This proves part (a). The proof of part (b) is almost identical: we consider the event that an edge is matched and replace y_e^* and t_u by x_e^* and 1 respectively and use the LP constraint (3). ■

Having Lemma 4.4, the proof of Theorem 4.3 is quite simple now.

Proof of Theorem 4.3. Given that an edge $e \in E$ is safe when considered, the expected profit from e is

$$w_e p_e y_e^* / \alpha = w_e x_e^* / \alpha.$$

Consider the point when $e = (u, v)$ is considered. By a union bound, we can see that

$$\begin{aligned} \Pr[e \text{ is safe}] &\geq 1 - \Pr[u \text{ loses patience}] - \Pr[v \text{ loses patience}] - \Pr[u \text{ is matched}] - \Pr[v \text{ is matched}] \\ &\geq 1 - 1/2\alpha - 1/2\alpha - 1/2\alpha - 1/2\alpha = 1 - 2/\alpha. \end{aligned}$$

Hence, the total expected profit is

$$\sum_e w_e \frac{x_e^*}{\alpha} \Pr[e \text{ is safe}] \geq \sum_e w_e x_e^* \frac{1 - 2/\alpha}{\alpha} \geq \frac{1 - 2/\alpha}{\alpha} \text{OPT}.$$

Plugging in $\alpha = 4$ gives an approximation ratio of 8, as desired. ■

4.4 Related Results and Variants

The problem was first formulated by Chen et al. [35]. They provided a 4-approximation for the unweighted version of the problem ($w_e = 1$ for all edges e). In fact, they showed that the greedy policy of choosing the edge e with highest p_e to probe achieves an approximation factor of 4. Their analysis is directly on the relations between the decision trees of the optimal policy and that of the greedy policy. Adamczyk [1] improved their analysis and showed that the greedy algorithm is in fact a 2-approximation for unweighted stochastic matching. Note that this is tight since the greedy algorithm is a 2-approximation even in the deterministic setting. However, this greedy algorithm (and other simple greedy schemes) can be seen to be arbitrarily bad for the weighted version. The above algorithm is from Bansal et al. [12]. They also provided a more nuanced 4-approximation (3-approximation for bipartite graphs) based on the *dependent rounding* technique [61]. Very recently, Adamczyk et al. [2] improved the ratio to 2.845 for bipartite graphs and Baveja et al. [15] obtained a 3.224-approximation for general graphs.

Open Question 2. *Is there a polynomial time approximation algorithm for the adaptive (unweighted or weighted) stochastic matching problem that can achieve an approximation ratio α with $\alpha < 2$?*

Online Stochastic Matching In an influential paper, Karp, Vazirani and Varirani [95] proposed the following *online matching* problem: There is a bipartite graph $G(U, V; E)$, where the vertices in U are given, and the vertices in V arrive one by one in an online fashion. When a new vertex $v \in V$ arrives, the edges incident on v are also revealed. We need to match v to an available vertex in U irrevocably, or leave v unmatched (in this case, we cannot match v later). Karp et al. gave a very simple and elegant randomized algorithm that achieves a competitive ratio of $1 - 1/e$, which is optimal if the order of arrivals of vertices in V are chosen by an *adversary*. An major motivation application of the online matching problem is *online advertising*. Here, each node in U represents an advertiser/bidder and each node in v represents an ad slot/keyword.

The *online stochastic matching* problem is a stochastic variant of the online matching problem, which was initially studied by Feldman et al. [53]. In this model, the bipartite graph is known,

but the sequence of arrivals are i.i.d. samples from a given distribution (instead of chosen by an adversary). Under some technical assumption, they gave a 0.67-competitive algorithm, beating the optimal $1 - 1/e$ -competitiveness known for adversarial arrivals [95]. Some improved bounds on this model were obtained [11, 113, 87]. The current best competitive ratio is 0.706, due to Jaillet and Lu [87]. and the best upper bound of competitive ratio of any online algorithm is 0.832, due to Manshadi et al. [113]. Goel and Mehta [64] considered a slight different model in which the arrival sequence is a random permutation (known as *random permutation model*). They showed a greedy algorithm achieves a $(1 - 1/e)$ competitive ratio. The ratio was improved to 0.653 by Karande et al. [92] and 0.696 by Mahdian and Yan [111]. The origin of the random permutation model is the popular *secretary* problem, which we review briefly in Section 9.

Mehta and Panigrahi [114] considered another related model where even the distribution is not given in advance, rather it arrives online. More precisely, when a new vertex v in V comes, we know a set of $\{p_{uv}\}_{u \in U}$ values. We have to assign v to an available vertex u in U , and then with probability p_{uv} the assignment is successful. If the assignment is not successful, we cannot assign v again, but u is still available for later arrivals. It is easy to see a simple greedy algorithm can achieve a competitive ratio of $1/2$. Mehta and Panigrahi improved the ratio to 0.567 for equal and vanishing probabilities (i.e., $p_{uv} = p$ for all $(u, v) \in E$ and $p \rightarrow 0$). They also showed that there is even no $(1 - 1/e)$ competitive algorithm for the problem. Recently, Mehta et al. [116] considered the more general case for unequal and vanishing probabilities, and presented a 0.534-competitive online algorithm, beating the trivial $1/2$ factor.

The Adword Problem and Online Stochastic LP: A closely related problem is the *Adwords* problem, which has important applications to sponsored search auctions. The problem is defined as follows. There are n advertisers and m keyword queries. The advertiser i has a budget B_i . In each time slot, a query j comes and we need to allocate query j to some advertiser $i \in [n]$. Let b_{ij} be the bid by advertiser i for query j . We use indicator variable x_{ij} to denote whether j is allocated to i . The revenue generated by the algorithm is $\sum_i \min(B_i, \sum_j b_{ij} x_{ij})$.

We can see that the problem reduces to the online matching problem if $B_i = 1$ and $b_{ij} = 0$ or 1 . The seminal paper by Mehta et al. [115] started the investigation of the problem and presented an optimal $(1 - 1/e)$ competitive algorithm in the worst case setting. Various stochastic settings (i.i.d. arrivals, random permutation models) have been studied extensively. The problem has been referred to as *online stochastic packing LP* in the literature (one can think the variables of an LP comes online, and the value of each new coming variable has to be determined immediately). An important quantity in the problem is the so-called budget-to-bid ratio $\gamma = \min_i B_i / \max_{ij} b_{ij}$. The ratio is usually very large in the online advertising applications, and hence it would be interesting to see if we can achieve better competitive ratios in this case. In fact, it is possible to achieve a competitive ratio of $(1 - \epsilon)$ provided that γ is large enough (in terms of ϵ, n, m). Agrawal, Wang and Ye [5] proved that γ should be at least $\Omega(\epsilon^{-2} \log m)$ in the random permutation model. After a series of papers [44, 52, 5], Kesselheim et al. [98] presented an optimal algorithm which achieves the condition established by Agrawal et al. In a recent elegant work, Agrawal and Devanur [4] generalized the result to a broader class of online stochastic *convex* optimization, using *Fenchel duality* and the *online mirror descent* algorithm developed in online learning. Gupta and Molinaro [77] considered a different generalization to both *packing and covering* constraints. They also made extensive use of the results from online learning.

5 Adaptive Stochastic Probing Problem

In this section, we consider the following general problem defined by Gupta and Nagarajan [78]. V is a set of elements and \mathcal{I} is a family of subsets of V . We say (V, \mathcal{I}) is a downwards-closed set systems if for any $I \in \mathcal{I}$, if $I' \subseteq I$, then $I' \in \mathcal{I}$. Many combinatorial constraints are downwards-closed, such as matroid, knapsack, matching and any intersection of them.

Problem 5.1. (*Stochastic Probing*) *The stochastic probing problem is defined over a universe V of elements with weights w_e for each $e \in V$. For each element $e \in V$, there is a probability p_e , indicating the probability that e is active. Each element is independent of all other elements. We are also given two downwards-closed set systems $(V, \mathcal{I}_{\text{out}})$ and $(V, \mathcal{I}_{\text{in}})$, which are called the outer packing constraint and the inner packing constraint respectively. We can adaptively probe the elements. If element e is probed and happens to be active (with probability p_e), we should choose e into our solution irrevocably. We require that the set Q of elements we can probe must be in \mathcal{I}_{out} and the set of chosen active elements must be in \mathcal{I}_{in} . Our goal is to design an adaptive policy which can choose a (random) set $S \subset V$ of active elements, the expected weight of S is maximized.*

The above problem naturally generalizes the stochastic matching problem: We can encode the patience level constraint by the outer packing constraint (\mathcal{I} is the set of all subgraphs which satisfies $\deg(v) \leq t_v$). The inner packing constraints dictates that the chosen set of edges form a matching (\mathcal{I} is the set of all matchings). The result in [78] applies to several important combinatorial packing constraints, such as matroid constraints, k -systems, k -column sparse packing integer programs. A key tool used here is an elegant abstraction of a class of LP rounding schemes, called contention-resolution schemes, introduced by Chekuri, Vondrák and Zenklusen [33].

5.1 Contention Resolution Schemes

A very popular and powerful methodology for designing approximation algorithms is to solve a linear programming (LP) relaxation first, and then round the fractional LP solution to an integral one. In the context of submodular maximization, Chekuri et al. [33] proposed a class of rounding schemes called Contention-Resolution schemes (CR schemes), which can be used to obtained constant approximations for the submodular maximization problem under several combinatorial constraints.

Now, we define what is a CR scheme. There are a set N of n elements and the set of feasible solutions can be captured by a downwards-closed set system $\mathcal{I} \subset 2^N$. Let $\mathcal{P}(\mathcal{I})$ be the polytope of the LP relaxation of the problem (so $\mathcal{I} \subseteq \mathcal{P}(\mathcal{I})$). We solve the LP to obtain a fractional optimal LP solution $x \in \mathcal{P}(\mathcal{I})$. We would like to round x to an integral near-optimal solution $\pi(x) \in \mathcal{I}$ via the following process π . For $0 \leq b \leq 1$ and $x \in \mathcal{P}(\mathcal{I})$, we use $R(bx) \subseteq N$ to denote the random set obtained by choosing each element $e \in N$ with probability bx_e .

Definition 5.2. (*CR Scheme*) *A (b, c) -balanced CR scheme π for a downwards-closed set system \mathcal{I} is a scheme such that for any $x \in \mathcal{P}(\mathcal{I})$, the scheme returns a set $\pi(I) \subseteq I = R(bx)$ with the following property:*

1. $\pi(I) \in \mathcal{I}$;
2. $\Pr[i \in \pi(I) \mid i \in I] \geq c$ for every element i .

A scheme is said to be monotone if $\Pr[i \in \pi(R_1)] \geq \Pr[i \in \pi(R_2)]$ whenever $i \in R_1 \subseteq R_2$. A scheme is said to be ordered if there is a permutation σ on N so that for any $I = R(bx) \subseteq N$, the output of the scheme $\pi(I)$ is the maximal independent subset of I obtained by considering elements in the order of σ .

From the above definition, we can see that each element e is selected by the CR scheme with probability at least bcx_e . Hence, if we want to maximize a linear function $\sum_e w_e x_e$ subject to $x \in \mathcal{I}$, we can immediately obtain a bc -approximation from a (b, c) -balanced CR scheme for \mathcal{I} . Monotone and ordered CR schemes will be particular useful to the stochastic probing problem.

We know efficient CR schemes for several important combinatorial constraints. For example, for any $0 < b \leq 1$, there is a $(b, (1 - e^{-b}/b))$ CR-schemes for matroids, a $(b, 1 - kb)$ ordered CR scheme for k -systems.¹⁰ See many more examples in [33].

Moreover, CR schemes have a very nice property that they can be combined for different constraints if they are all monotone. For example, if there is a monotone (b, c_1) -CR scheme π_1 for $\mathcal{P}_{\mathcal{I}_1}$ and a monotone (b, c_2) -CR scheme π_2 for $\mathcal{P}_{\mathcal{I}_2}$, one can combine them to obtain a monotone $(b, c_1 c_2)$ -CR scheme π for $\mathcal{P}_{\mathcal{I}_1 \cap \mathcal{I}_2}$, where $\pi(x) = \pi_1(x) \cap \pi_2(x)$. We can easily extend it to the intersection of more constraints as well.

5.2 Algorithm and Analysis for Stochastic Probing

Now, we show how a CR scheme enters the picture of the stochastic probing problem. In particular, we show the following theorem.

Theorem 5.3. *Consider an instance of the stochastic probing problem. Suppose the following hold:*

1. *There is a (b, c_{out}) -CR scheme π_{out} for $\mathcal{P}(\mathcal{I}_{\text{out}})$;*
2. *There is a monotone (b, c_{in}) ordered CR scheme π_{in} for $\mathcal{P}(\mathcal{I}_{\text{in}})$;*

Then, there is a polynomial time approximation algorithm which can achieve an approximation factor of $b(c_{\text{out}} + c_{\text{in}} - 1)$.

Given an instance of the stochastic probing problem with inner constraint $(V, \mathcal{I}_{\text{in}})$ and outer constraint $(V, \mathcal{I}_{\text{out}})$, we consider the following LP relaxation:

$$\begin{aligned} & \text{Maximize} && \sum_{e \in V} w_e x_e \\ & \text{s.t.} && x_e = p_e y_e, \quad \forall e \in V \\ & && x \in \mathcal{P}(\mathcal{I}_{\text{in}}) \\ & && y \in \mathcal{P}(\mathcal{I}_{\text{out}}) \end{aligned}$$

We assume that the above LP relaxation can be solved efficiently. First, we can easily obtain an analogue of Claim 4.2.

¹⁰In a k -system (V, \mathcal{I}) , for any $S \subseteq V$, every *maximal* independent set of S has size at least $1/k$ times the size of the *maximum* independent subset of S . Matroids are 1 systems, matchings on bipartite graphs are 2-systems, and the intersection of k -matroids are k -systems.

Claim 5.4. *The optimal value of LP is at least the optimal value OPT of the stochastic probing problem.*

Now, we present the algorithm here.

Algorithm 3: Stochastic Probing

- 1 Solve the LP relaxation and obtain the optimal LP solution (x_e, y_e) ;
 - 2 Pick $I \subset 2^V$ by choosing each $e \in V$ independently with probability by_e ;
 - 3 Let $P = \pi_{\text{out}}(I)$;
 - 4 Order elements in P according to the permutation given by the ordered CR scheme π_{in} ;
 - 5 **for** $i = 1 \rightarrow |P|$ **do**
 - 6 **if** $S \cup \{e_i\} \in \mathcal{I}_{\text{in}}$ **then**
 - 7 Probe e_i ;
 - 8 **If** e_i is active, let $S \leftarrow S \cup \{e_i\}$;
-

Now, we analyze the performance of the algorithm. We want to show the expected value of our solution $\mathbb{E}[w(S)]$ is large compared to the optimal LP value $\sum_e w_e x_e$. Let $J \subset V$ be the (random) set of active elements. Let $I \subset 2^V$ be the (random) set we picked in step 2, and P be the set we selected using π_{out} in step 3. From the algorithm, we can easily see that our solution $S = \pi_{\text{in}}(P \cap J)$

Lemma 5.5.

$$\Pr_{I, J, \pi_{\text{out}}, \pi_{\text{in}}} [e \in S] = \Pr_{I, J, \pi_{\text{out}}, \pi_{\text{in}}} [e \in \pi_{\text{in}}(\pi_{\text{out}}(I) \cap J)] \geq b(c_{\text{out}} + c_{\text{in}} - 1)x_e.$$

Proof. (sketch) First, we can see that $\Pr[e \in I \cap J] = by_e \cdot p_e = bx_e$ by the definition of I . By the definition of CR schemes, we have $\Pr[e \in P = \pi_{\text{out}}(I) \mid e \in I \cap J] \geq c_{\text{out}}$. As $P \subset I$, thus

$$\begin{aligned} \Pr[e \in \pi_{\text{in}}(P \cap J)] &= \Pr[e \in \pi_{\text{in}}(P \cap J) \wedge e \in I \cap P \cap J] \\ &= \Pr[e \in I \cap P \cap J] - \Pr[e \notin \pi_{\text{in}}(P \cap J) \wedge e \in I \cap P \cap J] \\ &= \Pr_{\pi} [e \in P \mid e \in I \cap J] \Pr_{I, J} [e \in I \cap J] - \Pr[e \notin \pi_{\text{in}}(P \cap J) \wedge e \in I \cap P \cap J] \\ &\geq bx_e \cdot c_{\text{out}} - \Pr[e \notin \pi_{\text{in}}(P \cap J) \wedge e \in I \cap P \cap J]. \end{aligned}$$

The remaining part is to show $K = \Pr[e \notin \pi_{\text{in}}(P \cap J) \wedge e \in I \cap P \cap J] \leq (1 - c_{\text{in}}) \cdot bx_e$. It is not hard to see that

$$K \leq \Pr[e \notin \pi_{\text{in}}(I \cap J) \wedge e \in I \cap J] = \Pr[e \notin \pi_{\text{in}}(I \cap J) \mid e \in I \cap J] \Pr[e \in I \cap J] \leq (1 - c_{\text{in}}) \cdot bx_e,$$

where the first inequality is due to the monotonicity of π_{in} and the final inequality is by the definition of the CR-schemes. ■

Now, it is straightforward to show Theorem 5.3. Indeed, we can see that

$$\mathbb{E}[w(S)] = \sum_e w_e \Pr[e \in S] \geq b(c_{\text{out}} + c_{\text{in}} - 1) \sum_e w_e x_e \geq b(c_{\text{out}} + c_{\text{in}} - 1) \text{OPT}.$$

5.3 Related Problems and Results

Bayesian online selection: A closely related model is the *Bayesian online selection problem (BOSP)*, defined as follows. We are given a set of elements V . Each element e is associated with a non-negative random variable X_e , with known distributions. The family of feasible solutions $\mathcal{I} \subseteq 2^V$ (encoded by some combinatorial constraint). We can adaptively choose to observe the elements one by one. Once we see the true value of X_e , we have to decide irrevocably whether to choose the element.¹¹ The goal is design a policy which chooses a feasible subset $S \in \mathcal{I}$, and maximizes the expected total value $\mathbb{E}[\sum_{e \in S} X_e]$. In some setting, the order is chosen by an adversary, instead of the policy. Chawla et al. [32] proposed a simple mechanism, called *sequential posted pricing* mechanism (SPM), for *Bayesian single-parameter auctions*. SPMs are closely related to, and in fact serve as important motivations for the stochastic probing problem and BOSP. See [83, 32, 6, 101, 108, 78, 55].

Prophet Inequality: In fact, the special case of BOSP where we can only choose one element was studied by Krenkel et al. back in 70s [103]. They provided a selection algorithm which returns a single element of expected value at least half of $\mathbb{E}[\max_{e \in V} X_e]$, i.e., half of the expected value obtained by an offline algorithm which knows the realization upfront. More precisely, let e be the element that their algorithm returns. It holds that

$$\mathbb{E}[X_e] \geq \frac{1}{2} \mathbb{E}[\max_{e \in V} X_e].$$

Such equalities are often called *the prophet inequalities*. Recently, Kleinberg and Weinberg [101] significantly generalized the above result for the general matroid constraints (the constraint of choosing one element is just a uniform matroid of rank 1).

Online CR-schemes: Recently, Feldman et al. [55] introduced an extension of CR schemes, called *online contention resolution schemes (OCRS)*. An OCRS is almost the same as the original CR scheme, except that the adversary chooses the order of the elements in I , and the OCRS needs to decide irrevocably whether to choose the element. They provide OCRS for a number of combinatorial constraints, such as matroids, matchings, knapsacks and their intersections, and applied to BOSP, stochastic probing and SPM. See their paper for more details.

6 Two Stage Stochastic Optimization - LP Approach

In this section, we consider two stage stochastic optimization models. In particular, we study the *two stage recourse* models: In the first stage, only distributional information is available, and one commits on some initial actions. In the second stage, the actual data is revealed and we can take additional actions (called recourse actions) to augment the initial solution to a feasible solution. The cost for the recourse action can be more expensive than the initial action. The stochastic set cover problem (Problem 1.3) is a typical example in this model.

Let us recall the notations. Let U be the set of ground elements and \mathcal{S} the weighted family of subsets of U . The actual demand (i.e., the set of elements we need to cover) follows a distribution \mathcal{D} . We can select $\mathcal{S}^I \subseteq \mathcal{S}$ as the first stage action. In the second stage, the actual demand set

¹¹We observe the true value of X_e before we make the decision. This is a key difference from the stochastic probing problem.

$A \subseteq U$ ($A \sim \mathcal{D}$) is revealed. We can select some extra subsets $\mathcal{S}_A^{II} \subseteq \mathcal{S}$ as the recourse to cover the remaining elements of A . The *inflation factor* is $\lambda > 1$, that is a set in the second stage is λ times more expensive than in the first stage. We call a possible demand set $A \subseteq U$ a *scenario*. Let p_A be the probability of scenario A . The goal is to minimize $w(\mathcal{S}^I) + \mathbb{E}_A[w(\mathcal{S}_A^{II})]$.

We use the following LP relaxation. x_S denotes whether the set S is chosen in the first-stage, and $y_{A,S}$ denotes whether the set S is chosen in scenario A .

$$\begin{aligned} & \text{Minimize } \sum_{S \in \mathcal{S}} w_S x_S + \lambda \sum_{A, S \in \mathcal{S}} p_A w_S y_{A,S} \\ & \text{s.t. } \sum_{S: e \in S} x_S + \sum_{S: e \in S} y_{A,S} \geq 1 && \forall A, e \in A \\ & x_S, y_{A,S} \geq 0 && \forall S, A \end{aligned}$$

The above LP is usually called a *stochastic LP*. The first constraint indicates that the first and second stage actions combined is a set cover for any scenario A . If there are only a polynomial number of scenarios and all $\{p_A\}$ are given explicitly, we can solve the stochastic LP in polynomial time. However, in most applications, there can be an exponential number of scenarios, given in an implicit manner. For example, each element is in A with certain probability independent of others. A more general and more challenging model is the so-called *black-box model*, in which we are only given a procedure from which we can sample the scenarios from \mathcal{D} . We can not even write down the stochastic LP explicitly for the black-box model. Nevertheless, Shmoys and Swamy [125] showed that there is a polynomial time approximation scheme for solving the LP by taking at most polynomial number of samples from the black-box, if λ is bounded by a polynomial of the input size. Their algorithm is an adaptation of the ellipsoid algorithm. Later, they presented another approximation schemes based on the *sample average approximation (SAA) method*. SAA is a very natural approach for stochastic optimization. We simply sample N scenarios from the black-box. Then, we solve sample-average problem, which is a deterministic problem. Shmoys and Swamy [130] showed that we only need a polynomial number of samples, under some mild conditions. Charikar et al. [31] provided an alternative and somewhat simpler proof for the same result. In fact, their approach is more general: it does not only hold for stochastic LP, it can also be used for the computational intractable problems, where only an approximation algorithm for the deterministic sample-average problem is known.

Now, we come back to the stochastic set cover problem. We assume the polynomial-scenario model. For the black-box model, we only pay an extra factor of $(1 + \epsilon)$ for any constant $\epsilon > 0$ based on the above discussion.

Theorem 6.1. *There is a $2(\ln n + 1)$ factor approximation algorithm, where n is the number of elements.*

Proof. We first consider the standard LP relaxation for set cover with universe U .

$$\text{minimize } \sum_{S \in \mathcal{S}} w_S x_S \quad \text{subject to} \quad \sum_{S: e \in S} x_S \geq 1 \quad \forall e \in U; \quad x_S \geq 0 \quad \forall S \in \mathcal{S}.$$

Let the optimal LP value be $\text{LP}(U)$. It is long known that a standard independent rounding algorithm can find an integral set cover with cost at most $O(\log n \text{LP}(U))$ with high probability. Using the exponential clock algorithm by Buchbinder et al. [25], one can get an integral set cover with cost at most $(\ln n + 1)\text{LP}(U)$, with probability 1. Denote this algorithm by \mathcal{A} . Let $\alpha = \ln n + 1$.

For any element e and scenario A , we can see that either $\sum_{S:e \in S} x_S \geq 1/2$ or $\sum_{S:e \in S} y_{A,S} \geq 1/2$. Consider $E = \{e : \sum_{S:e \in S} x_S \geq 1/2\}$. $\{2x_S\}$ is a feasible fractional solution for the set cover instance with universe E . Running algorithm \mathcal{A} , we get an integral solution covering E with cost at most $\alpha \text{LP}(E) \leq \alpha \sum_S 2x_S w_S$. In the second stage, suppose the scenario is A . $\{2y_{A,S}\}$ is a feasible fractional solution covering all elements in $A \setminus E$. Hence, we can get an integral solution covering $A \setminus E$ with cost at most $\alpha \text{LP}(A \setminus E) \leq \alpha \lambda \sum_S 2y_{A,S} w_S$. So, the overall expected cost is at most $\alpha \sum_S 2x_S w_S + \alpha \lambda \sum_A p_A \sum_S 2y_{A,S} w_S$. Thus we have a 2α -approximation algorithm for stochastic 2-stage set cover problem. \blacksquare

7 Two Stage Stochastic Optimization- Boosted Sampling

In this section, we introduce another classical technique for dealing with two stage stochastic models, called boosted sampling, introduced by Gupta et al. [80]. We use the minimum rooted Steiner tree problem as an example to illustrate the technique.

Problem 7.1. (*Two Stage Stochastic Rooted Steiner Tree*) *In the deterministic rooted Steiner tree problem, we are given an edge-weighted graph $G(V, E)$ with edge weight $w : E \rightarrow \mathbb{R}^+$ satisfying triangle inequality and a root $r \in V$. We are also given a set of terminals $S \subseteq V$, and we want to choose a tree of the minimum total weight to connect all terminals and the root r . In the two stage stochastic model, there is a probability distribution \mathcal{D} on the set of terminals. A set of edge E_0 may be bought in the first stage, under cost $\{w_e\}_{e \in E}$. In the second stage, the real set $S \sim \mathcal{D}$ of terminals is revealed and we may need to buy some extra edges E_S so that $E_0 \cup E_S$ is a feasible solution. However, we need to pay σw_e for edge e in the second stage, where $\sigma > 1$ is the inflation factor (which is also an input). The goal again is to minimize the expected cost of the solution, that is $w(E_0) + \mathbb{E}_S[\sigma w(E_S)]$.*

7.1 Cost Sharing Functions

We will make a crucial use of cost-sharing functions in the analysis of the algorithm. Consider the deterministic Steiner tree problem. There is a client in each terminal. We would like to figure out a way to divide the cost of the Steiner tree among the client set S . Now, we formally define what is a cost-sharing function. Suppose there is an α -approximation algorithm \mathcal{A} that solves the Steiner tree problem. Moreover, we require existence of a polynomial time algorithm $\text{Aug}_{\mathcal{A}}$ (called *augmentation algorithm*) that can augment a solution $\mathcal{A}(S)$ to a feasible solution for S' for $S \subset S'$.

Definition 7.2. *The function $\xi : 2^V \times V \rightarrow \mathcal{R}$ is a β -strict cost sharing function if the following properties hold:*

1. (*Positivity*) For a set $S \subset V$, $\xi(S, j) > 0$ only for $j \in S$.
2. (*Fairness*) For a set $S \subset V$, $\sum_{j \in S} \xi(S, j) \leq w(\text{OPT}(S))$.
3. (*Strictness*) If $S' = S \cup T$ for $S \cap T = \emptyset$, then $\sum_{j \in T} \xi(S', j) \geq \frac{1}{\beta} \times \text{cost of augmenting the solution } \mathcal{A}(S) \text{ to a solution of } S' \text{ using } \text{Aug}_{\mathcal{A}}$.

Define $\xi(S, A) = \sum_{j \in A} \xi(S, j)$.

In some cases, $\text{Aug}_{\mathcal{A}}$ can be obtained by zeroing out the costs of elements picked in $\mathcal{A}(S)$ and run \mathcal{A} again. In this case, we typically have $\beta = \alpha$. In general case, β is usually larger because it is usually easier to find a better approximation algorithm than to find a better augment algorithm. Now, we provide a cost sharing function for the Steiner problem, which will be useful later.

Theorem 7.3. *There is a 2-approximation algorithm \mathcal{A} with a 2-strict cost sharing function ξ for the rooted minimum Steiner tree problem.*

Proof. Given a set of terminals S , \mathcal{A} compute a minimum spanning tree on $S \cup \{r\}$. It is well known that the cost of $\mathcal{A}(S)$ is within 2 times the cost of the optimal minimum Steiner tree of S .

With \mathcal{A} , now we determine the value of $\xi(S, j)$ according to \mathcal{A} . For any set S and vertex j , if $j \in S$, set $\xi(S, j) = w_{(p_j, j)}/2$ where p_j is the parent of j in the spanning tree $\mathcal{A}(S)$; if $j \notin S$, set $\xi(S, j) = 0$. Clearly, $\sum_{j \in S} \xi(S, j) = \sum_{j \in S} w_{(p_j, j)}/2 = w(\mathcal{A}(S))/2 \leq w(\text{OPT}(S))$. So we have shown the fairness.

To show the strictness, we first define $\text{Aug}_{\mathcal{A}}$. Let $S' = S \cup T$. $\text{Aug}_{\mathcal{A}}$ first zeros out the edges in $\mathcal{A}(S)$ and then run Prim's algorithm. The solution obviously include $\mathcal{A}(S)$. For each terminal $j \in T$, let p_j be its parent in $\mathcal{A}(S \cup T)$. Consider the execution of Prim, edge (j, p_j) is added by $\text{Aug}_{\mathcal{A}}$ as well. So the cost of augmenting the solution $\mathcal{A}(S)$ to a solution for S' is exactly $\sum_{j \in T} w_{(j, p_j)}$. We can also see that $\sum_{j \in T} \xi(S', j) = 1/2 \sum_{j \in T} w_{(j, p_j)}$, which shows $\beta = 2$. ■

We can in fact use the approximation algorithm proposed in [26] and the same cost sharing function as above.

Theorem 7.4. *There is a 1.39-approximation algorithm, along with a 2-strict cost sharing function for the rooted minimum Steiner tree problem.*

If the problem is that you are given two sets S, T separately and required to obtain a solution for $S \cup T$, you can first compute an α -approximation solution $\phi(S)$ for S , then augment it to a solution of $S \cup T$, $\phi(S \cup T)$, using $\text{Aug}_{\mathcal{A}}$. As

$$w(\phi(S \cup T)) - w(\phi(S)) \leq \beta \xi(S \cup T, T) \leq \beta \xi(S \cup T, S \cup T) \leq \beta w(\text{OPT}(S \cup T))$$

and

$$w(\phi(S)) \leq \alpha w(\text{OPT}(S)) \leq \alpha w(\text{OPT}(S \cup T)),$$

$\phi(S \cup T)$ is a $(\alpha + \beta)$ -approximation solution for $S \cup T$.

7.2 Stochastic Steiner Trees

Now, let us come back to the two stage stochastic Steiner tree problem. Consider the following algorithm.

Algorithm 4: Boost-and-Sample

- 1 Draw $\lfloor \sigma \rfloor$ independent sample scenarios $D_1, \dots, D_{\lfloor \sigma \rfloor}$ from the distribution \mathcal{D} . Let $D = \cup_i D_i$;
 - 2 Run the algorithm \mathcal{A} to construct an α -approximation F_0 for terminals D as the first stage solution;
 - 3 In second stage, when S is realized, run $\text{Aug}_{\mathcal{A}}$ to compute F_S such that $F_0 \cup F_S$ is a feasible Steiner tree for S ;
-

Now, we analyze the performance of the above algorithm.

Theorem 7.5. *Let \mathcal{A} be an α -approximation algorithm for minimum rooted Steiner tree problem that admits a β -strict cost-sharing function. Then Boost-and-Sample is an $(\alpha + \beta)$ -approximation algorithm for the 2-stage stochastic Steiner tree problem.*

Proof. Let F_0^* be the first-stage component of the optimal solution, and F_S^* be the second-stage component for scenario S . Hence the optimal cost is

$$Z^* = w(F_0^*) + \sigma \mathbb{E}_S[w(F_S^*)].$$

Now consider the cost of Boost-and-Sample. Without loss of generality, assume σ is an integer. We use $\text{Sols}(S)$ to denote the set of feasible solutions for terminal set S .

Consider the first stage. Define $\hat{F}_1 = F_0^* \cup F_{D_1}^* \cup \dots \cup F_{D_\sigma}^*$. Thus $\hat{F}_1 \in \text{Sols}(D)$. Moreover, we can see that

$$\mathbb{E}_D[w(\hat{F}_1)] \leq w(F_0^*) + \sum_i \mathbb{E}_D[w(F_{D_i}^*)] = w(F_0^*) + \sigma \mathbb{E}_{D_i}[w(F_{D_i}^*)] = Z^*.$$

Since \mathcal{A} is an α -approximation algorithm, the expected cost of our first stage solution satisfies $\mathbb{E}_D[w(F_0)] \leq \alpha \mathbb{E}_D[w(\hat{F}_1)] \leq \alpha Z^*$.

Next consider the second stage, which is slightly more trickier. Note that S follows the same distribution as any D_i . The key is to consider an alternate process to generate the sets D_i and S : Draw $\sigma + 1$ scenarios $\hat{D}_1, \dots, \hat{D}_{\sigma+1}$ from the distribution \mathcal{D} , choose a random K uniformly from $\{1, 2, \dots, \sigma + 1\}$, and set $S = \hat{D}_K$ and $D = \cup_{i \neq K} \hat{D}_i$. Now let $\hat{D} = \cup_{i=1}^{\sigma+1} \hat{D}_i$ and $\hat{D}_{-i} = \cup_{l \neq i} \hat{D}_l$. Intuitively, D has σ copies of D_i , but \hat{D} has $\sigma + 1$ copies. So $\mathbb{E}_{\hat{D}}[w(\text{OPT}(\hat{D}))]$ is not much larger than $\mathbb{E}_{\hat{D}}[w(\text{OPT}(D))]$. Moreover, S is simply one of the $\sigma + 1$ copies. The cost for connecting S should be only $1/(\sigma + 1)$ of $w(\text{OPT}(\hat{D}))$. We provide the details below.

Recall F_S is the set of edges we add in the second stage. According to the β -strictness, we can see $w(F_S) \leq \beta \xi(D \cup S, S \setminus D)$. By the fairness, we can see that $\sum_{i=1}^{\sigma+1} \xi(\hat{D}, \hat{D}_i \setminus \hat{D}_{-i}) \leq w(\text{OPT}(\hat{D}))$. From the process, we can see $\xi(D \cup S, S \setminus D)$ is just a random term on the left hand side of the above inequality. Hence, we obtain that

$$\mathbb{E}_{D,S}[\xi(D \cup S, S \setminus D)] \leq \frac{1}{\sigma + 1} \mathbb{E}_{\hat{D}}[w(\text{OPT}(\hat{D}))].$$

Moreover, by the sub-additivity, it holds that

$$\mathbb{E}_{\hat{D}}[w(\text{OPT}(\hat{D}))] \leq w(F_0^*) + \sum_{i=1}^{\sigma+1} \mathbb{E}[w(F_{\hat{D}_i}^*)] \leq \frac{\sigma + 1}{\sigma} Z^*.$$

Combining the above inequalities, we get $\mathbb{E}_S[\sigma w(F_S)] \leq \beta Z^*$. Overall, the total expected cost is at most $(\alpha + \beta)Z^*$. ■

7.3 Related Results

Using the boosted sampling technique, Gupta et al. [80] provided constant factor approximation algorithms for several two stage stochastic optimization problems, including Steiner tree, vertex cover, facility location and Steiner network. Note that their framework works even if \mathcal{D} is given by the black-box model. However, if we assume the nodes are independent, they showed that it is possible to obtain improved approximation algorithms.

problem	black-box model	independence model
Steiner Tree (rooted, CU)	3.55 [81]	3.55 [81]
Steiner Tree (unrooted, CU)	6 [81, 57]	5[81, 57]
Steiner Tree (unrooted, N)	$O(\log^6 n \log \log n)$ [73]	$O(\log^6 n \log \log n)$ [73]
Steiner Forest (CU)	2976 [81, 76]	5 [81, 57]
Vertex Cover (Ar)	$2 + \varepsilon$ [127]	$2 + \varepsilon$ [127]
Facility Location (Ar)	$2.369 + \varepsilon$ [127]	$2.369 + \varepsilon$ [127]
Set Cover (Ar)	$(1 + o(1)) \ln n$ [127]	$(1 + o(1)) \ln n$ [127]

Table 1: Best known approximation ratios for two stage stochastic optimization problems with recourse. The objective function is the total expected cost. Correlated Uniform (CU): All the elements share a common inflation factor λ_A in each scenario A . Nonuniform (N): Each element e may have its own inflation factor λ_e , independent of different scenarios. Arbitrary (Ar): The inflation factor $\lambda_{e,A}$ of element e may depend on both e and scenario A . In all settings, the maximum inflation factor is at most a polynomial.

The work has generated a number of followups. We mention a very small subset here. Gupta and Pál [81] studied the unrooted version of the two stage stochastic Steiner tree problem, and provided the first constant factor approximation algorithm. The problem is exactly the same as we defined before, except that there is no root r . This case is technically more challenging since the problem is not sub-additive any more.

Gupta and Kumar [75] generalized the above result by providing the first constant factor approximation algorithm for the two stage stochastic Steiner forest problem,¹² using a primal-dual LP approach. In fact, the cost sharing function in the last section is closely related to the dual variable. In a very recent work, Gupta and Kumar [76] showed a simple greedy algorithm for Steiner Forest problem with an approximation ratio of 96 with 2880-strict cost sharing function, which implies an 2976 approximation for the two stage stochastic Steiner forest problem.

In another generalization, Gupta et al. [72] consider the two stage stochastic Steiner tree problem with non-uniform inflation factors, i.e., the cost of each edge in the second stage may be inflated by a different factor. They provided the first poly-logarithmic approximation factor and showed an approximation hardness of $\Omega(\log \log n)$.

7.4 Other Two Stage or Multi-Stage Models

Ravi and Sinha [121] and Immorlica et al. [86] initiated the study on approximation algorithms for two stage stochastic combinatorial optimization with recourse. They provided approximation algorithms for several problems for the polynomial scenario and independent-activation settings. Later, various techniques, such as boosted sampling, stochastic LP, SAA, were developed for handling the black box model. We refer interested reader to [129] for an early survey. In Table 1, we list the current best known approximation ratios for the two stage stochastic models of several other classic combinatorial problems. Note that here the objective is to minimize the expected overall cost.

Many of the above results have been generalized to k stages for arbitrary constant k . See e.g.,

¹²In a Steiner forest instance, we are given an edge weighted undirected graph G and a set of vertex pairs $\{(s_i, t_i)\}_{i \in [m]}$. We need to choose a subset T of edges such that s_i and t_i are connected for all $i \in [m]$.

[130, 81, 127]. There are also several closely related models proposed in the literature. We review some of them here.

Two Stage Robust Optimization: Robust optimization takes a more conservative viewpoint in decision making under uncertainty. The goal in robust optimization is to minimize the worst case cost over all possible scenarios, instead of minimizing the average cost over scenarios, as done in stochastic optimization problems. The setting of a two stage robust optimization problem is almost the same as the two stage stochastic optimization counterpart, except that there is no probability distribution over the scenarios, and the objective is to minimize $w(x) + \max_A f_A(x, y_A)$. Here x is the first stage action and y_A is the recourse action for scenario A . Such model and its variations have also been studied extensively in the last decade. We refer interested readers to the following papers and the references therein [45, 65, 51, 7, 99, 79].

Two Stage Risk-Averse Stochastic Optimization: The problem setting is exactly the same as Section 6, except that the goal is to minimize the expected total cost subject to the constraint

$$\Pr_A[2\text{nd stage cost for scenario } A > \theta] \leq \rho.$$

Here, θ and ρ are input parameters as well. Such constraints are called *threshold probability constraints*, or *chance constraints*, and used to model risk averse behaviors of the decision maker.

Such chance constraints can be captured using similar LP-relaxation as well. For the black-box model, Swamy [128] proposed an FPTAS for solving such LP and showed how to round the LP for several combinatorial problems.

8 Fixed Set Problems and Stochastic Knapsack - Poisson Approximation

In this section, we introduce the Poisson approximation technique and apply it the fixed set stochastic optimization problems, and the adaptive stochastic knapsack problems. The technique is very useful to handle the distribution of the sum of independent random variables (not necessarily identical), which appears quite frequently in various stochastic optimization problems.

We first illustrate the technique by considering the fixed set stochastic shortest path problem (Problem 1.1). The result can be generalized to a wide class of fixed set problems (see [108]).

Recall each edge e has a non-negative random length μ_e . The objective is to find an s - t path P such that $\Pr[\sum_{e \in P} \mu_e \leq \mathbb{C}]$, is maximized. Without loss of generality, we can assume $\mathbb{C} = 1$ and all μ_e s take values from $[0, 1]$. For ease of notation, for a set S of edges, we write $\mu(S) = \sum_{e \in S} \mu_e$. Let S^* denote the optimal feasible set and $\text{OPT} = \Pr[\mu(S^*) \leq 1]$ the optimal value.

Theorem 8.1. *For any $\epsilon > 0$, there is a polynomial time approximation algorithm for finds an s - t path S such that*

$$\Pr[\mu(S) \leq 1 + \epsilon] \geq \Pr[\mu(S^*) \leq 1] - \epsilon,$$

where S^* is the optimal path.

We start by a lemma saying that if the $\Pr[\mu(S) \leq 1]$ is not negligible, $\mathbb{E}[\mu(S)]$ can not be very large.

Lemma 8.2. *Suppose each edge e has a non-negative random weight μ_e taking values from $[0, 1]$. Then, for any set S of edges, and $\frac{1}{2} > \epsilon > 0$, if $\Pr[\mu(S) \leq 1] \geq \epsilon$, then $\mathbb{E}[\mu(S)] \leq 3/\epsilon$.*

Intuitively, if $\mathbb{E}[\mu(S)]$ is very large, $\mu(S)$ should be large with high probability (hence $\Pr[\mu(S) \leq 1]$ should be very small). The proof is not difficult and can be found in [108].

If $\text{OPT} \leq \epsilon$, then there is nothing to do since any feasible solution achieves the desired approximation guarantee. Hence, we focus on the other case where $\text{OPT} > \epsilon$. We call an edge e *heavy edge* if $\mathbb{E}[\mu_e] > \epsilon^{10}$. Otherwise we call it *light*. By Lemma 8.2, we can see that the number of heavy edges in S^* is at most $\frac{3}{\epsilon^{11}}$.

Enumerating Heavy Elements: We enumerate all possible set of heavy edges with size at most $3/\epsilon^{11}$. There are at most $n^{3/\epsilon^{11}}$ such possibilities. Suppose we successfully guess the set of heavy edges in S^* . In the following parts, we mainly consider the question that given a set H of heavy edges, how to choose a set L of light edges such that their union S is a feasible solution, and $\Pr[\mu(S) \leq 1 + \epsilon]$ is close to optimal.

Dealing with Light Elements: Unlike heavy edges, there could be many light edges in S^* . Handling such edges involves two technique. The first is the discretization, which transforms each distribution to one with a constant size support in $[0, 1]$.

The second key ingredient is the Poisson approximation technique. We apply a theorem of Le Cam [105], which shows that the distribution of the sum of the discretized weights of light edges is very close to a compound Poisson distribution, which can be completely determined by a constant dimensional vector (which we call the signature of L).

Then, we enumerate all possible signatures (there are only polynomial number of them), and checking whether there is an s - t path $S = L \cup H$ (where H is the set of heavy edges we enumerate, and L is the set of light edges in S), such that the signature of L is the enumerated signature.

8.1 Discretization

We discuss how to discretize the size distributions for edges, using parameter ϵ .

We say that edge e realizes to a “large” size if $\mu_e > \epsilon^4$. Otherwise we say that e realizes to a “small” size. We use $\tilde{\mu}_e$ to denote the size after discretization and $\tilde{\pi}_e$ its distribution. The discretization consists of following two steps.

1. Small size region: In the small size region, $\tilde{\mu}_e$ follows a Bernoulli distribution, taking only values 0 and ϵ^4 . The probability values $\Pr[\tilde{\mu}_e = 0]$ and $\Pr[\tilde{\mu}_e = \epsilon^4]$ are set such that $\mathbb{E}[\tilde{\mu}_e \mid \mu_e \leq \epsilon^4] = \mathbb{E}[\mu_e \mid \mu_e \leq \epsilon^4]$.
2. Large size region: If μ_e realizes to a large size, we simply discretize it as follows: Let $\tilde{\mu}_e = \lfloor \frac{\mu_e}{\epsilon^5} \rfloor \epsilon^5$ (i.e., we round a large size down to a multiple of ϵ^5).

We denote the set of the discretized sizes by $\mathcal{S} = \{s_0, s_1, \dots, s_{z-1}\}$ where $s_0 = 0, s_1 = \epsilon^5, s_2 = 2\epsilon^5, s_3 = 3\epsilon^5, \dots, s_{z-1}$. Note that $s_1 = \epsilon^5, \dots, s_{1/\epsilon-1} = \epsilon^4 - \epsilon^5$ are also included in \mathcal{S} , even though their probability is 0. It is straightforward to see that $|\mathcal{S}| = z = O(1/\epsilon^5)$. This finishes the description of the discretization.

It is not difficult to show the behavior of the sum of their discretized distributions is very close to that of their original distributions.¹³ The proof of the following lemma is completely standard and omitted here.

¹³As $\epsilon \rightarrow 0$, the discretized distribution clearly converges to the original distribution in the weak topology (e.g., in Lévy-Prokhorov metric or transportation metric, see [48]).

Lemma 8.3. *Let S be a set of edges such that $\mathbb{E}[\mu(S)] \leq 3/\epsilon$. We have that*

1. $\Pr[\mu(S) \leq 1] \leq \Pr[\tilde{\mu}(S) \leq 1 + \epsilon] + O(\epsilon)$;
2. $\Pr[\tilde{\mu}(S) \leq 1] \leq \Pr[\mu(S) \leq 1 + \epsilon] + O(\epsilon)$.

8.2 Poisson Approximation

We use $\tilde{\pi}_e$ to denote the distribution of the discretized edge weight $\tilde{\mu}_e$, i.e., $\tilde{\pi}_e(s_i) = \Pr[\tilde{\mu}_e = s_i]$. For an edge e , we define its *signature* to be the vector

$$\mathbf{Sg}(e) = (\tilde{\pi}_e(s_1), \tilde{\pi}_e(s_2), \tilde{\pi}_e(s_3), \dots, \tilde{\pi}_e(s_{z-1})),$$

where $\tilde{\pi}_e(s) = \lfloor \tilde{\pi}_e(s) \cdot \frac{n}{\epsilon^6} \rfloor \cdot \frac{\epsilon^6}{n}$ for all nonzero discretized size $s \in \mathcal{S} \setminus \{0\} = \{s_1, s_2, \dots, s_{z-1}\}$. For a set S of edges, its *signature* is defined to be the sum of the signatures of all edges in S , i.e.,

$$\mathbf{Sg}(S) = \sum_{e \in S} \mathbf{Sg}(e).$$

We use $\mathbf{Sg}(S)_k$ to denote the k th coordinate of $\mathbf{Sg}(S)$. By Lemma 8.2, $\sum_{k=1}^{z-1} \mathbf{Sg}(S)_k \cdot s_k = \sum_{k=1}^{z-1} \mathbf{Sg}(S)_k \cdot s_k \leq 3/\epsilon$. Thus $\mathbf{Sg}(S)_k \leq 3/\epsilon^5$ for all k . Therefore, the number of possible signatures is bounded by $(3n/\epsilon^{11})^{|\mathcal{S}|-1}$, which is polynomial in n .

We will show shortly that it suffices to enumerate all possible signatures for light edges. For this purpose, we need a notation to measure the difference between two distributions, called the *total variation distance* (also call *statistical distance*), defined as follows (for discrete distributions):

$$\Delta(X, Y) = \sum_k \left| \Pr[X = k] - \Pr[Y = k] \right|.$$

Obviously, if $\Delta(X, Y) = 0$, two distributions are identical. The following lemma shows that if the signatures of two sets are the same, the total variation distance between their distributions is very small.

Lemma 8.4. *Let S_1, S_2 be two sets of light edges such that $\mathbf{Sg}(S_1) = \mathbf{Sg}(S_2)$ and $\mathbb{E}[\tilde{X}(S_1)] \leq 3/\epsilon, \mathbb{E}[\tilde{X}(S_2)] \leq 3/\epsilon$. Then, the total variation distance between $X(S_1)$ and $X(S_2)$ satisfies*

$$\Delta(\tilde{X}(S_1), \tilde{X}(S_2)) = \sum_s \left| \Pr[\tilde{X}(S_1) = s] - \Pr[\tilde{X}(S_2) = s] \right| = O(\epsilon).$$

The following Poisson approximation theorem by Le Cam [105], rephrased in our language, is essential for proving Lemma 8.4. Suppose we are given a K -dimensional vector $V = (V_1, \dots, V_K)$. Let $\lambda = \sum_{i=1}^K V_i$. we say a random variable Y follows the *compound Poisson distribution corresponding to V* if it is distributed as $Y = \sum_{j=1}^N Y_j$ where N follows Poisson distribution with expected value λ :

$$\Pr(N=k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad \text{for } k \geq 0,$$

(denoted as $N \sim \text{Pois}(\lambda)$) and Y_1, \dots, Y_N are i.i.d. random variables with $\Pr[Y_j = 0] = 0$ and $\Pr[Y_j = k] = V_k/\lambda$ for $k \in \{1, \dots, K\}$ and $j \in \{1, \dots, N\}$.

Lemma 8.5. [105] Let X_1, X_2, \dots be independent random variables taking integer values in $\{0, 1, \dots, K\}$, let $X = \sum X_i$. Let $\pi_i = \Pr[X_i \neq 0]$ and $V = (V_1, \dots, V_K)$ where $V_k = \sum_i \Pr[X_i = k]$. Suppose $\lambda = \sum_i \pi_i = \sum_k V_k < \infty$. Let Y be the compound Poisson distribution corresponding to vector V . Then, the total variation distance between X and Y can be bounded as follows:

$$\Delta(X, Y) = \sum_{k \geq 0} \left| \Pr[X = k] - \Pr[Y = k] \right| \leq 2 \sum_i \pi_i^2.$$

Proof of Lemma 8.4. For an edge b , we let $\bar{\mu}_e$ be the random variable that $\Pr[\bar{\mu}_e = s] = \bar{\pi}_e(s)$ for $s = s_1, s_2, \dots, s_{z-1}$, and $\bar{\mu}_e = 0$ with the rest of the probability mass. Similarly, we use $\tilde{\mu}(S)$ to denote $\sum_{b \in S} \bar{\mu}_e$ for a set S of edges. By definition of $\bar{\mu}_e$, we have that $\Delta(\bar{\mu}_e, \tilde{\mu}_e) \leq \epsilon/n$ for any edge b . Since S_1 and S_2 contains at most n edges, we can show that

$$\Delta(\bar{X}(S_1), \tilde{X}(S_1)) \leq \epsilon \quad \text{and} \quad \Delta(\bar{X}(S_2), \tilde{X}(S_2)) \leq \epsilon.$$

First, for any S such that $\mathbb{E}[\tilde{X}(S)] \leq 3/\epsilon$, we can see that $\sum_{b \in S} \Pr[\tilde{\mu}_e \neq 0] \leq \mathbb{E}[\tilde{\mu}(S)]/\epsilon^4 \leq 3/\epsilon^5$. If we apply Lemma 8.5 to both $\bar{X}(S_1)$ and $\bar{X}(S_2)$, we can see they both correspond to the same compound Poisson distribution, say Y , since their signatures are the same. Moreover, since the total variation distance is a metric, we have that

$$\begin{aligned} \Delta(\tilde{\mu}(S_1), \tilde{\mu}(S_2)) &\leq \Delta(\tilde{\mu}(S_1), \bar{\mu}(S_1)) + \Delta(\bar{\mu}(S_1), Y) + \Delta(Y, \bar{\mu}(S_2)) + \Delta(\bar{\mu}(S_2), \tilde{\mu}(S_2)) \\ &\leq 2\epsilon + 2 \sum_{b \in S_1} \left(\Pr[\bar{\mu}_e \neq 0] \right)^2 + 2 \sum_{b \in S_2} \left(\Pr[\bar{\mu}_e \neq 0] \right)^2 + \epsilon = O(\epsilon). \end{aligned}$$

This finishes the proof of the lemma. ■

8.3 Algorithm for Fixed Set Problems

Now, we present our approximation algorithm for the fixed set stochastic shortest path problem.

Algorithm 5: Fixed Set Stochastic Shortest Path

- 1 Discretize the size distributions of for all light edges;
 - 2 Enumerate all possible heavy edge sets H with $\mathbb{E}[\tilde{X}(H)] < 3/\epsilon$;
 - 3 **for each such H do**
 - 4 Enumerate all possible signatures \mathbf{Sg} ;
 - 5 **for each such \mathbf{Sg} do**
 - 6 Try to find an s - t path $S = H \cup L$, such that $\mathbf{Sg}(L) = \mathbf{Sg}$;
 - 7 Pick the path S with the largest $\Pr[\mu(S) \leq 1 + \epsilon]$;
-

In step (a), we can use the pseudopolynomial time algorithm for the exact version of the shortest path problem to find a set L with the signature exact equal to \mathbf{Sg} . This can be done by standard dynamic programming.¹⁴ Since \mathbf{Sg} is a vector with $O(\epsilon^{-5})$ coordinates and the value of each coordinate is bounded by $O(n)$, it can be encoded by an integer which is at most $n^{\text{poly}(1/\epsilon)}$. Thus the pseudopolynomial time algorithm actually runs in $\text{poly}(n, n^{\text{poly}(1/\epsilon)})$ time, which is a

¹⁴We allow non-simple paths. The exact version of simple path is NP-hard, since it generalizes the Hamiltonian path problem.

polynomial. We also need to compute the value $\Pr[\mu(S) \leq 1 + \epsilon]$. The problem is in fact #P-hard [100]. However, we can use the FPTAS developed in [107] to obtain an $(1 \pm \epsilon)$ -approximated estimation, which suffices for our purpose. There are a polynomial number of combinations of heavy edges. So it is not hard to see the algorithm runs in $n^{\text{poly}(1/\epsilon)}$ time overall.

It is fairly straightforward to show that the algorithm achieves the guarantee stated in Theorem 8.1. Suppose we have guessed the heavy edges H^* of the optimal solution S^* , and the signature of the light edges of $S^* = H^* \cup L^*$. Suppose the path we found is $S = H^* \cup L$. Using 8.3, we can see that the original distribution $\mu(L)$ ($\mu(L^*)$ resp.) is close to the discretized distribution $\tilde{\mu}(L)$ ($\tilde{\mu}(L)$ resp.). By Lemma 8.4, $\tilde{\mu}(L)$ is close to the $\tilde{\mu}(L^*)$. Hence, we can see that the distribution of $\mu(L)$ is close to that $\mu(L^*)$. Hence, S behaves very similar to S^* .

Astute readers may have realized that we did not use much special properties about the combinatorics of the shortest path problem except a pseudopolynomial time algorithm that solves the exact version. In fact, we can generalize the above result to all combinatorial optimization problems which admit a pseudopolynomial time algorithm for the exact version, including spanning tree, k-median on trees, knapsack. Moreover, instead of maximizing $\Pr[\mu(S) \leq 1]$, we can consider the more general expected utility maximization problem. See the detailed results in [108, 106].

8.4 Poisson Approximation For Adaptive Stochastic Knapsack

The Poisson approximation technique can also be used to obtain bi-criterion PTAS for the adaptive stochastic knapsack problem and its generalizations. This is somewhat surprising since it is not even clear at first sight where to apply the Poisson approximation technique, which can only handle the sum of a fixed set of random variables.

We need an important notion, called *block-adaptive policies*, introduced by Bhargat et al. [20]. In a block-adaptive policy, instead of inserting the items one at a time, we insert a subset (a.k.a., block) of items at a time. In terms of the decision tree of a policy, each node in the tree corresponding to the insertion of a block of items.

A remarkable property proved in [20] is that there exists a block-adaptive policy, which corresponds to a decision tree with only $O(1)$ nodes, that can approximate the optimal policy, modulo an ϵ fraction of profit and an ϵ fraction of knapsack capacity. The result is further generalized to stochastic knapsack with arbitrary subset constraints [108], which allows us to handle cancellation of jobs¹⁵ and precedence constraints. Hence, it suffices to consider only block-adaptive policies. Since there are only constant number of nodes in the decision tree, we can enumerate all different topologies of the decision tree. Now, for each node of the tree (which corresponds to a block of items), we can use Poisson approximation technique to approximate the distribution of the sum of random variables in the block. More concretely, after fixing the topology of the decision tree, we can enumerate the signatures of all blocks in polynomial time. Then, we can use a dynamic program to search a block-adaptive policy, which matches the enumerated signature. In the analysis, we need to show that if two block-adaptive policies have the same tree topology and same signature for every node of the tree, two policies must behave similarly (in particular, obtain almost the same expected profit).

Using the above idea, we can obtain a policy which can produce a profit at least $(1 - \epsilon)\text{OPT}$ using $(1 + \epsilon)\text{C}$ capacity (i.e., a bi-criterion PTAS), for the adaptive stochastic knapsack problem,

¹⁵In this generalization, called stochastic knapsack with cancellations, we can think each item as a job and its (random) size and as the length of the job. We can cancel the job before it is finished. We do not get any profit from the canceled job.

even when job cancellation is allowed. Even though the above idea is clean, the details are quite technical, and we refer interested readers to [108].

8.5 Related Work

Expected Utility Maximization and Threshold Probability Maximization:

In Section 8, we considered the fixed set stochastic shortest path problem, where our goal is to maximize the threshold probability $\Pr[\mu(S) \leq 1]$. This is a very special case of the following more general expected utility maximization problem.

Problem 8.6. (*Expected Utility Maximization*) Suppose we have n elements, each having a non-negative random weight μ_e . We would like to choose a subset S , subject to some combinatorial constraint. Consider a utility function $\mathcal{U} : \mathbb{R}^+ \rightarrow \mathbb{R}^+$. If the cost of the solution is x , we obtain a utility value $\mathcal{U}(x)$. The goal is to find a solution S which maximizes the expected utility $\mathbb{E}[\mathcal{U}(\mu(S))]$, where $\mu(S)$ is the (random) cost of the solution S , i.e., $\mu(S) = \sum_{e \in S} \mu_e$.

Consider the utility function: $\mathcal{U}(x) = 1$ for $x \in [0, 1]$ and $\mathcal{U}(x) = 0$ for $x > 1$. It is easy to see that $\mathbb{E}[\mathcal{U}(\mu(S))] = \Pr[\mu(S) \leq 1]$. Hence, maximizing the expected utility is equivalent to maximizing the threshold probability. The expected utility is known to be very versatile in expressing diverse risk-averse or risk-prone behaviors. For example, we can use an increasing concave utility function to capture the risk-averse behaviors. The theory was axiomatized by von Neumann and Morgenstern in 1940s [132, 56] (known as von Neumann- Morgenstern expected utility theory in the economics literature).

Li and Deshpande [106] first studied the expected utility maximization problem in the context of stochastic combinatorial optimization. They considered general combinatorial constraints and utility functions. Specifically, they can handle the class of combinatorial constraints, which admit pseudopolynomial time algorithms. Examples include spanning tree, matching, simple path, matroid, knapsack. We can also obtain Theorem 8.1 as a corollary.

Their approach is very different. The high level idea is very simple. We first observe that for the exponential utility function $\mathcal{U}(x) = \alpha^x$ for any $\alpha \in \mathbb{C}$, the problem essentially reduces to a deterministic optimization problem. Indeed, fix an arbitrary solution S and a number α . Due to the independence of the elements, we can see that $\mathbb{E}[\mathcal{U}(\mu(S))] = \mathbb{E}[\alpha^{\mu(S)}] = \mathbb{E}[\alpha^{\sum_{e \in S} \mu_e}] = \mathbb{E}[\prod_{e \in S} \alpha^{\mu_e}] = \prod_{e \in S} \mathbb{E}[\alpha^{\mu_e}]$. Now, we consider a more general utility function \mathcal{U} . For simplicity, we assume \mathcal{U} is essentially supported in a bounded interval. Then, for utility function \mathcal{U} , we try to use a short exponential sum to approximate \mathcal{U} . More concretely, we want an exponential sum $\mathcal{C}(x) = \sum_{i=1}^L c_i \phi_i^x$ with L being a constant, which satisfies $|\mathcal{U}(x) - \mathcal{C}(x)| \leq \epsilon, \forall x \geq 0$. If we can do so, instead of directly optimizing $\mathbb{E}[\mathcal{U}(\mu(S))]$, we can optimize $\mathbb{E}[\mathcal{C}(\mu(S))]$, which can be done using dynamic programming. The remaining thing is to approximate a function by a short exponential sum. In general, this is not possible for an arbitrary function over the entire \mathbb{R} . However, we only need to do it for a bounded interval. Then, we can utilize a theorem by Jackson in approximation theory, combined with some other tricks, to find such an approximation. See [106] for the details.

For increasing concave utility functions, Bhargat and Khanna [21] obtained the first PTAS, if the exact version of the deterministic counterpart can be solved in pseudopolynomial time. They proved a utility equivalence theorem and showed that it suffices to enumerate only a polynomial number of “configurations”. In full version of [106], the authors reproduced the same result for increasing concave utility functions, using the function approximation paradigm we just described.

Fixed Set Stochastic Shortest Path: Nikolova et al. [119] studied the fixed set stochastic shortest path for Gaussian, Poisson and exponential distributions. For Gaussian distributed edges, it is easy to see that the length of a path is also Gaussian distributed, with the mean/variance being the sum of the means/variances of individual edges. Now, we can view the problem as a two-dimensional optimization problem, one for the mean, and the other for the variance. Imagine the mean-variance plane. Each point in the plane correspond to the (mean, variance) pair of a path. Consider the convex hull H of these points (they call it the *path polytope*). It is not hard to show that maximizing the threshold probability is equivalent to maximizing

$$(1 - \sum_{i \in S} \mathbb{E}[\mu_i]) / \sqrt{\sum_{i \in S} \text{Var}[\mu_i]},$$

which is quasi-convex on the path polytope. As a consequence, the quantity, thus the threshold probability, is maximized at a vertex of the path polytope (under certain technical condition). The vertices of H can be enumerated in $O(n^{\log n})$ time due to a result in parametric shortest path by Carstensen [28]. Hence, by enumerating the vertices of H , they obtained an exact $O(n^{\log n})$ time algorithm for maximizing the probability that the length of the path is at most 1, i.e., $\Pr(w(S) \leq 1)$, assuming all edges are normally distributed and there is a path with its mean at most 1. Later, Nikolova [118] extended the result to an FPTAS for any problem under the same Gaussian assumptions, if the deterministic version of the problem can be solved in polynomial time.

Fixed Set Stochastic Knapsack: In the fixed set stochastic knapsack problem, we are given a knapsack of capacity 1, and a set of items, each with a random size s_i and a deterministic profit v_i , and an overflow probability γ . We are asked to pick a subset S of items such that

$$\Pr\left(\sum_{i \in S} s_i \geq 1\right) \leq \gamma$$

and the total profit $\sum_{i \in S} v_i$ is maximized.

Kleinberg et al. [100] first considered the fixed set stochastic knapsack problem with Bernoulli-type distributions and provided a polynomial-time $O(\log 1/\gamma)$ approximation. For item sizes with exponential distributions, Goel and Indyk [63] provided a bi-criterion PTAS, and for Bernoulli-distributed items they gave a quasi-polynomial approximation scheme. Goyal and Ravi [66] showed a PTAS for Gaussian distributed sizes. Bhalgat et al.[20] applied the discretization technique to both adaptive stochastic knapsack and fixed set stochastic knapsack. For the later, they provide a bi-criterion PTAS: for any constant $\epsilon > 0$, there is a polynomial time algorithm that can find a solution S with the profit as least the optimum and $\Pr(\sum_{i \in S} x_i \geq 1 + \epsilon) \leq \gamma + \epsilon$. Using the result for expected utility maximization, the same result can be also obtained by the utility function approximation approach [106] or the Poisson approximation approach [108], with somewhat simpler proofs and better running times.

For both fixed set stochastic shortest path and fixed set stochastic knapsack, the current best approximations are bi-criterion additive PTASes. So we have the following obvious open questions.

Open Question 3. S^* is the optimal solution for the fixed set stochastic shortest path problem. For any $\epsilon > 0$, whether there is a polynomial time approximation algorithm that finds an s - t path S , such that $\Pr[\mu(S) \leq 1] \geq \Pr[\mu(S^*) \leq 1] - \epsilon$,

Open Question 4. S^* is the optimal solution for the fixed set stochastic knapsack problem. For any $\epsilon > 0$, whether there is a polynomial time approximation algorithm that finds a set of item S with the profit as least the optimum and $\Pr(\sum_{i \in S} x_i \geq 1) \leq \gamma + \epsilon$.

We finally note that a recent result by Daskalakis et al. [39] is closely related to the above problems. Their problem can be stated in the following abstract form: Given a random vector X generated by a known product distribution over $\{0, 1\}^n$ and a threshold value $0 \leq \theta \leq 1$, output a non-negative vector $w \in \mathbb{R}^n$, with $\|w\|_1 = 1$, which maximizes $\Pr[w \cdot X \geq \theta]$. They provided an additive PTAS under certain technical condition. Their technique borrows ideas from the study of *linear threshold functions* in complexity theory and makes use of the *Berry-Esseen theorem* (a quantitative version of the central limit theorem). Removing their technical condition is also an interesting open problem.

9 Other Stochastic Models

9.1 Stochastic Universal Approximation

Suppose we want to distribute a file from the source s to a set T of nodes in a network G . If we know T , then we can just compute a steiner tree connecting $T \cup \{s\}$. Karger and Minkoff [93] considered the *maybecast problem* which is another stochastic version of the Steiner tree problem,

Problem 9.1. (*Maybecast Problem*) Each node i chooses to contact the source (we say i is active) with probability p_i independently. We need to fix a path P_i to s for each node i . If the node i is active, all edges in P_i become active. Our goal is to minimize the expected total number of active edges.¹⁶

Karger and Minkoff [93] showed that the shortest path tree heuristic can be very bad ($\Omega(\sqrt{n})$ factor worse than the optimum) and also obtained a constant factor approximation algorithm for the problem by reducing the problem to the r -gathering problem, which is a variant of facility location problem with capacity lower bound for each open facility.

The maybecast problem is closely related to the notion of the universal approximation introduced in [89]. We take the *universal steiner tree* problem for example. In this problem, we still need to fix a path P_i to s for each node i . However, we do not assume each node becomes active in a probabilistic manner, Instead, we take a worst case (or *robust approximation*) perspective, by bounding the approximation ratio for any subset of S . More precisely, we want to minimize

$$\max_{S \subseteq V} \frac{\text{cost}(\cup_{i \in S} P_i)}{\text{cost}(\text{OPT}(S))},$$

where $\text{OPT}(S)$ is the optimal Steiner connecting $S \cup \{s\}$. Since the line of research does not involve any stochasticity, we do not go into the details and refer interested readers to [89, 82] and the references therein.

¹⁶If we use multicast to distribute the file to all active nodes, the number of active edges is proportional to the total transmission cost.

9.2 Secretary Problem

The secretary problem is a classical stochastic sequential decision problem, introduced by Dynkin in 1960s [49]. The basic version is a very simple to state, yet the result is stunning at first glance, which makes the problem quite popular even in public media. Suppose we would like to hire the best secretary out of n applicants. The applicants are interviewed one by one in random order. After the interview of each applicant, a decision whether to hire the applicant must be made immediately and irrevocably. During the interview, we know the rank of the applicant among all applicants interviewed so far. The goal is to maximize the chance that we hire the best one. There is very simple strategy that can guarantee the probability is at least $1/e$, irrespective of how large n is. It works as follows: It first interviews the first n/e applicants without hiring any of them. Then, it hires the first one who is the best one among the applicants interviewed so far. The ratio $1/e$ is tight for large n .

Kleinberg [102] studied a generalization of the problem, in which we want to select k candidates and maximize their sum. He provided an algorithm that can achieve a competitive ratio of $1 - O(\sqrt{1/k})$, which is asymptotically optimal. Babaioff et al. [10] studied a significant generalization, called the *matroid secretary problem*, in which the selected set must be an independent set in a given matroid. The problem has attracted a lot of attentions (see e.g., [85, 126, 30, 47, 104, 54]). For several special types of matroids, constant approximations are known. However, for a general matroid, the current best approximation is $O(\log \log r)$ [104, 54], where r is the rank of the given matroid.

Open Question 5. *Is there a constant factor approximation algorithm for the matroid secretary problem?*

The secretary problem has important applications in mechanism design, and played a similar role as the prophet inequalities we mentioned in Section 5.3. However, they are very different from the technical perspective. For example, the aforementioned matroid prophet inequality has a 2-approximation [101].

9.3 Stochastic Multi-armed Bandit

Multi-armed Bandit problems nowadays refer to many different variants which are too large to survey. They are mostly sequential decision making problem featured with an exploration and exploitation trade-off. The most basic version is the following problem: We are given n arms. The i -th arm is associated with an unknown reward distribution supported on $[0, 1]$ with mean θ_i . If we pull the arm, we get an i.i.d. sample from the distribution. There are T rounds. In each round, we can choose one arm to pull. Our goal is to minimize the *regret*, which is defined to be the difference between the total reward obtained by our algorithm over the T round, and the reward we can obtain if we keep playing the best arm. For this basic problem and its numerous extensions, we can obtain an $o(T)$ regret.¹⁷ Instead of playing one arm, we may play a combinatorial set of arms in each round. This is the *combinatorial bandit* problem. There are numerous other extensions and variations. We refer interested readers to [29, 23] for more comprehensive treatments.

¹⁷There are several variations of the definition of regret. For the basic version we mention here, we can achieve an $O(\sqrt{T})$ regret. For some other variant (such as pseudo-regret), it is possible to achieve better guarantee (such as logarithmic regret).

Markovian bandit: Another extremely important class of stochastic multi-armed bandit problems we have not mentioned yet is the *Markovian* bandit problems. Here, the reward distribution of each arm follows from a Markov chain. Suppose the chains and the transition probabilities are known. Upon a play of an arm, the state of the arm changes according to the transition probabilities of the corresponding Markov chain. We would like to design an adaptive policy to play the arms and maximize the expected reward. A celebrated result in this domain is *Gittin's index* [62], due to John Gittins, who obtained a polynomial time solution for maximizing the expected discounted reward. In fact, Gittins's solution is a very simple-looking greedy solution, which assigns each state of the Markov chain a fixed number (the Gittin's index), and always play the arm with the current largest index. Several simpler proofs of the result are discovered later (see e.g., [58]) and covered in several text books. Bertsimas and Niño-Mora [18] provided a unified view of several problems that admit similar efficient greedy-like algorithms, via the notion of *extended polymatroid*. However, many variants are PSPACE-hard, shown by Papadimitriou and Tsitsiklis [120]. There is also a body of work studying approximation algorithms with provable approximation factors for these problems. See [68, 70, 67, 71, 110] and the references therein. In fact, some bandit models generalizes the stochastic knapsack we considered in the beginning (see e.g., [71, 110]).

Bandit Arm Selection: Now, we briefly review some problems and results related to Problem 1.4 we introduced in Section 1. Such problems are also called *pure exploration multi-armed bandit* problems. The most basic version is the best arm identification problem, in which the goal is to select the single best arm. Bechhofer [17] first formulated the problem for Gaussian arms in 1954. There has been a resurgence of interest for the problem in the last decade [50, 112, 59, 91, 94, 88, 34]. Mannor and Tsitsiklis [112] showed that for any algorithm that returns the correct answer with probability at least $1 - \delta$, it requires $\Omega\left(\sum_{i=2}^n \Delta_i^{-2} \ln \delta^{-1}\right)$ samples in expectation for any instance, where Δ_i is the difference between the mean of the best arm and that of the i th arm. Chen and Li [34] obtained the current best upper bound

$$O\left(\Delta_2^{-2} \ln \ln \Delta_2^{-1} + \sum_{i=2}^n \Delta_i^{-2} \ln \delta^{-1} + \sum_{i=2}^n \Delta_i^{-2} \ln \ln \min(n, \Delta_i^{-1})\right).$$

The above bound is worst case optimal, since there is a matching worst-case lower bound for each of the three terms. In fact, the first term is nearly instance optimal¹⁸ (see [34] for the details), hence not improvable. The second term is instance optimal due to the lower bound in [112]. Only the third term is worst-case optimal.

Open Question 6. *Obtain a nearly instance optimal algorithm for the best arm identification problem. We conjecture that the optimal bound is of the form $\Delta_2^{-2} \ln \ln \Delta_2^{-1} + L(I)$, where $L(I)$ is an instance-wise lower bound.*

The worst case sample complexity in the PAC (Probably Approximately Correct) setting is also well studied. In the PAC setting, the algorithm should return an arm whose mean is at most ϵ worse than the optimal one, with probability at least $1 - \delta$. There is a matching (worst case) lower and upper bound $\Omega(n \ln \delta^{-1} / \epsilon^2)$ [112, 50]. The generalization to selecting the top- k arms has also been studied extensively for the last few years (see e.g., [59, 60, 134, 91, 90, 24, 97, 96, 27]). Recently, Chen et al. [36] initiated the study of the combinatorial pure exploration problem, which generalizes the cardinality constraint to more general combinatorial constraints (e.g., matroid).

¹⁸Instance optimality means that the upper and lower bounds match for every instance. Worst-case optimality means that the upper and lower bounds match for an infinite class of instances.

9.4 Clustering Stochastic Points

There are two popular models for stochastic points in a metric space. In the existential uncertainty model, each node v is presented at a fixed point with probability p_v , independent of other point. In the locational uncertainty model, the location of a node follows some given distribution. Cormode and McGregor [37] considered the k -center clustering problem¹⁹ for the locational model in a finite metric graph, and gave a bi-criterion constant approximation. The result was improved later to a true constant factor approximation by Guha and Munagala [69]. Munteanu [117] studied the 1-center problem (a.k.a. the minimum enclosing ball problem) for stochastic points in fixed dimensional Euclidean space and provided a PTAS. Recently, Huang et al. [84] obtained a PTAS for the more general j -flat center problem (i.e., the center is a j -flat, i.e., a j -dimensional affine subspace) for stochastic point, using an extension of the powerful geometric notation ϵ -kernel coresets, introduced by Agarwal et al. [3], to the stochastic setting. *In the (k, j) -projective clustering problem, we are asked to choose k j -flat, such that the maximum distance for any point to its closest j -flat is minimized.* Extending the above results to other clustering problems (e.g., the k -center problem \mathbb{R}^d with $k = O(1)$, $d = O(1)$) is an interesting future direction.

10 Concluding Remarks and Acknowledgement

Due to the limit of space and the authors' knowledge, this survey is by no means comprehensive. We apologize in advance for the omission of any important results. We would like to thank Xiaodong Hu for inviting us to write the survey. We would like to thank Haotian Jiang, Lijie Chen, Lingxiao Huang for many useful suggestions. Jian Li and Yu Liu are supported in part by the National Basic Research Program of China grants 2015CB358700, 2011CBA00300, 2011CBA00301, and the National NSFC grants 61202009, 61033001, 61361136003.

References

- [1] Marek Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011.
- [2] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. Improved approximation algorithms for stochastic matching. *arXiv preprint arXiv:1505.01439*, 2015.
- [3] P.K. Agarwal, S. Har-Peled, and K.R. Varadarajan. Approximating extent measures of points. *Journal of the ACM*, 51(4):606–635, 2004.
- [4] Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424. SIAM, 2015.
- [5] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operation Research*, 2014.

¹⁹The goal here is to choose k points in the metric space as centers, such that the expected maximum assignment cost of any node is minimized. There are two versions. In the fixed assignment version, we have to fix the assignment of a node to a center in our solution. In the free assignment version, a node can be assigned to its closest center, depending on its realized location.

- [6] S. Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *FOCS*, pages 512–521, 2011.
- [7] Barbara Anthony, Vineet Goyal, Anupam Gupta, and Viswanath Nagarajan. A plant location guide for the unsure: Approximation algorithms for min-max location problems. *Mathematics of Operations Research*, 35(1):79–101, 2010.
- [8] D. Assaf. Renewal decisions when category life distributions are of phase-type. *Mathematics of Operations Research*, pages 557–567, 1982.
- [9] D. Assaf and B. Levikson. On optimal replacement policies. *Management Science*, pages 1304–1312, 1982.
- [10] M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443, 2007.
- [11] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *Proceedings of the 18th European Symposium on Algorithms, LNCS 6346*, pages 170–181. Springer, 2010.
- [12] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
- [13] Nikhil Bansal and Viswanath Nagarajan. On the adaptivity gap of stochastic orienteering. In *Integer Programming and Combinatorial Optimization*, pages 114–125. Springer, 2014.
- [14] J.F. Bard and J.E. Bennett. Arc reduction and path preference in stochastic acyclic networks. *Management Science*, 37(2):198–215, 1991.
- [15] Alok Baveja, Amit Chavan, Andrei Nikiforov, Aravind Srinivasan, and Pan Xu. Improved bounds in stochastic matching and optimization. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 40. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
- [16] B.C.Dean, M.X.Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. In *Proc. 45th FOCS*, 2004.
- [17] Robert E Bechhofer. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *The Annals of Mathematical Statistics*, pages 16–39, 1954.
- [18] Dimitris Bertsimas and José Niño-Mora. Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems. *Mathematics of Operations Research*, 21(2):257–306, 1996.
- [19] Anand Bhalgat. A $(2 + \epsilon)$ -approximation algorithm for the stochastic knapsack problem. 2012.
- [20] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1647–1665. SIAM, 2011.

- [21] Anand Bhalgat and Sanjeev Khanna. A utility equivalence theorem for concave functions. In *Integer Programming and Combinatorial Optimization*, pages 126–137. Springer, 2014.
- [22] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
- [23] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.
- [24] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. *arXiv preprint arXiv:1205.3181*, 2012.
- [25] Niv Buchbinder, Joseph Seffi Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 535–544. ACM, 2013.
- [26] Jaroslav Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved lp-based approximation for steiner tree. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 583–592. ACM, 2010.
- [27] Wei Cao, Jian Li, Yufei Tao, and Zhize Li. On top-k selection in multi-armed bandits and hidden bipartite graphs. In *Neural Information Processing Systems*, 2015.
- [28] Patricia J Carstensen. Complexity of some parametric integer and network programming problems. *Mathematical Programming*, 26(1):64–75, 1983.
- [29] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- [30] S. Chakraborty and O. Lachish. Improved competitive ratio for the matroid secretary problem. In *SODA*, pages 1702–1712, 2012.
- [31] Moses Charikar, Chandra Chekuri, and Martin Pál. Sampling bounds for stochastic optimization. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 257–269. Springer, 2005.
- [32] S. Chawla, J.D. Hartline, D.L. Malec, and B. Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, pages 311–320. ACM, 2010.
- [33] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
- [34] Lijie Chen and Jian Li. On the optimal sample complexity for best arm identification. *arXiv preprint arXiv:1511.03774*, 2015.
- [35] Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming, LNCS 5555*, pages 266–278, 2009.

- [36] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 379–387, 2014.
- [37] Graham Cormode and Andrew McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–200. ACM, 2008.
- [38] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3):197–206, 1955.
- [39] Constantinos Daskalakis, Anindya De, Ilias Diakonikolas, Ankur Moitra, and Rocco A Servedio. A polynomial-time approximation scheme for fault-tolerant distributed storage. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 628–644. SIAM, 2014.
- [40] Brian C Dean, Michel X Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 395–404. Society for Industrial and Applied Mathematics, 2005.
- [41] Brian C Dean, Michel X Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.
- [42] C. Derman, G.J. Lieberman, and S.M. Ross. A renewal decision problem. *Management Science*, pages 554–561, 1978.
- [43] C. Derman and DR Smith. Renewal decision problem-random horizon. *Mathematics of Operations Research*, pages 225–232, 1979.
- [44] Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78. ACM, 2009.
- [45] Kedar Dhamdhere, Vineet Goyal, R Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 367–376. IEEE, 2005.
- [46] Kedar Dhamdhere, R. Ravi, and Mohit Singh. On two-stage stochastic minimum spanning trees. In *Proceedings of the 11th International Conference on Integer Programming and Combinatorial Optimization, IPCO’05*, pages 321–334, Berlin, Heidelberg, 2005. Springer-Verlag.
- [47] Michael Dinitz and Guy Kortsarz. Matroid secretary for regular and decomposable matroids. *SIAM Journal on Computing*, 43(5):1807–1830, 2014.
- [48] Richard M Dudley. *Real analysis and probability*, volume 74. Cambridge University Press, 2002.
- [49] Eugene B Dynkin. The optimum choice of the instant for stopping a markov process. In *Soviet Math. Dokl*, volume 4, 1963.

- [50] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *Computational Learning Theory*, pages 255–270. Springer, 2002.
- [51] Uriel Feige, Kamal Jain, Mohammad Mahdian, and Vahab Mirrokni. Robust combinatorial optimization with exponential scenarios. In *Integer Programming and Combinatorial Optimization*, pages 439–453. Springer, 2007.
- [52] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *Algorithms-ESA 2010*, pages 182–194. Springer, 2010.
- [53] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S Muthukrishnan. Online stochastic matching: Beating $1-1/e$. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 117–126. IEEE, 2009.
- [54] Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $o(\log \log(\text{rank}))$ -competitive algorithm for the matroid secretary problem. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1189–1201. SIAM, 2015.
- [55] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *SODA*, 2016.
- [56] P.C. Fishburn. *The foundations of expected utility*. Dordrecht: Reidel, 1982.
- [57] Lisa Fleischer, Jochen Könemann, Stefano Leonardi, and Guido Schäfer. Strict cost sharing schemes for steiner forest. *SIAM J. Comput.*, 39(8):3616–3632, October 2010.
- [58] Esther Frostig and Gideon Weiss. Four proofs of gittins multiarmed bandit theorem. *Annals of Operations Research*, pages 1–39, 1999.
- [59] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2012.
- [60] Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck. Multi-bandit best arm identification. In *Advances in Neural Information Processing Systems*, pages 2222–2230, 2011.
- [61] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):360, 2006.
- [62] John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979.
- [63] Ashish Goel and Piotr Indyk. Stochastic load balancing and related problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 579–586. IEEE, 1999.
- [64] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 982–991. Society for Industrial and Applied Mathematics, 2008.

- [65] Daniel Golovin, Vineet Goyal, and R Ravi. Pay today for a rainy day: improved approximation algorithms for demand-robust min-cut and shortest path problems. In *STACS 2006*, pages 206–217. Springer, 2006.
- [66] V. Goyal and R. Ravi. A ptas for the chance-constrained knapsack problem with random item sizes. *Operations Research Letters*, 38(3):161–164, 2010.
- [67] S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. *ICALP*, pages 496–507, 2009.
- [68] Sudipto Guha and Kamesh Munagala. Approximation algorithms for partial-information based stochastic control with markovian rewards. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, 2007.
- [69] Sudipto Guha and Kamesh Munagala. Exceeding expectations and clustering uncertain data. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 269–278. ACM, 2009.
- [70] Sudipto Guha, Kamesh Munagala, and Peng Shi. Approximation algorithms for restless bandit problems. *Journal of the ACM (JACM)*, 58(1):3, 2010.
- [71] A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. In *FOCS*, pages 827–836, 2011.
- [72] Anupam Gupta, MohammadTaghi Hajiaghayi, and Amit Kumar. Stochastic steiner tree with non-uniform inflation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 134–148. Springer, 2007.
- [73] Anupam Gupta, Mohammadtaghi Hajiaghayi, and Amit Kumar. Stochastic steiner tree with non-uniform inflation. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '07/RANDOM '07, pages 134–148, Berlin, Heidelberg, 2007. Springer-Verlag.
- [74] Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R Ravi. Approximation algorithms for stochastic orienteering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1522–1538. SIAM, 2012.
- [75] Anupam Gupta and Amit Kumar. A constant-factor approximation for stochastic steiner forest. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 659–668. ACM, 2009.
- [76] Anupam Gupta and Amit Kumar. Greedy algorithms for steiner forest. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 871–878, New York, NY, USA, 2015. ACM.
- [77] Anupam Gupta and Marco Molinaro. How experts can solve lps online. In *Algorithms-ESA 2014*, pages 517–529. Springer, 2014.
- [78] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization*, pages 205–216. Springer, 2013.

- [79] Anupam Gupta, Viswanath Nagarajan, and R Ravi. Thresholded covering algorithms for robust and max–min optimization. *Mathematical Programming*, 146(1-2):583–615, 2014.
- [80] Anupam Gupta, Martin Pál, R Ravi, and Amitabh Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 417–426. ACM, 2004.
- [81] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. *SIAM J. Comput.*, 40(5):1361–1401, September 2011.
- [82] Mohammad T Hajiaghayi, Robert Kleinberg, and Tom Leighton. Improved lower and upper bounds for universal tsp in planar metrics. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 649–658. Society for Industrial and Applied Mathematics, 2006.
- [83] M.T. Hajiaghayi, R. Kleinberg, and T. Sandholm. Automated online mechanism design and prophet inequalities. In *National Conference on Artificial Intelligence*, volume 22, page 58, 2007.
- [84] Lingxiao Huang, Jian Li, Jeff M Phillips, and Haitao Wang. ϵ -kernel coresets for stochastic points. *arXiv preprint arXiv:1411.0194*, 2014.
- [85] S. Im and Y. Wang. Secretary problems: Laminar matroid and interval scheduling. In *SODA*, pages 1265–1274, 2011.
- [86] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 691–700. Society for Industrial and Applied Mathematics, 2004.
- [87] Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2013.
- [88] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil’ucb: An optimal exploration algorithm for multi-armed bandits. *COLT*, 2014.
- [89] Lujun Jia, Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, and Ravi Sundaram. Universal approximations for tsp, steiner tree, and set cover. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 386–395. ACM, 2005.
- [90] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 511–518, 2010.
- [91] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 655–662, 2012.

- [92] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 587–596. ACM, 2011.
- [93] DR Karger and Maria Minkoff. Building steiner trees with incomplete global knowledge. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 613–623. IEEE, 2000.
- [94] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1238–1246, 2013.
- [95] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.
- [96] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *arXiv preprint arXiv:1407.4443*, 2014.
- [97] Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251, 2013.
- [98] Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 303–312. ACM, 2014.
- [99] Rohit Khandekar, Guy Kortsarz, Vahab Mirrokni, and Mohammad R Salavatipour. Two-stage robust network design with exponential scenarios. *Algorithmica*, 65(2):391–408, 2013.
- [100] Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.
- [101] R. Kleinberg and S.M. Weinberg. Matroid prophet inequalities. In *STOC*, pages 123–136, 2012.
- [102] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 630–631. Society for Industrial and Applied Mathematics, 2005.
- [103] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts and processes with finite value. *Advances in Prob*, 4:197–266, 1978.
- [104] Oded Lachish. $O(\log \log \text{rank})$ competitive ratio for the matroid secretary problem. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 326–335. IEEE, 2014.
- [105] L. Le Cam. An approximation theorem for the poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181–1197, 1960.

- [106] Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 797–806. IEEE, 2011.
- [107] Jian Li and Tianlin Shi. A fully polynomial-time approximation scheme for approximating a sum of random variables. *Operations Research Letters*, 42(3):197–202, 2014.
- [108] Jian Li and Wen Yuan. Stochastic combinatorial optimization via poisson approximation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013.
- [109] R.P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.
- [110] Will Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1154–1163. SIAM, 2014.
- [111] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 597–606. ACM, 2011.
- [112] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648, 2004.
- [113] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011.
- [114] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 728–737. IEEE, 2012.
- [115] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.
- [116] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1388–1404. SIAM, 2015.
- [117] Alexander Munteanu, Christian Sohler, and Dan Feldman. Smallest enclosing ball for probabilistic data. In *Proceedings of the thirtieth annual symposium on Computational geometry*, page 214. ACM, 2014.
- [118] E. Nikolova. Approximation Algorithms for Reliable Stochastic Combinatorial Optimization. *APPROX*, pages 338–351, 2010.
- [119] Evdokia Nikolova, Jonathan A Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *Algorithms–ESA 2006*, pages 552–563. Springer, 2006.

- [120] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
- [121] R Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. *Mathematical Programming*, 108(1):97–114, 2006.
- [122] A.E. Roth, T. Sonmez, and U. Unver. Kidney exchange. *Quarterly Journal of Economics*, 119:457–488, 2004.
- [123] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, Berlin, 2003.
- [124] Alexander Shapiro, Darinka Dentcheva, et al. *Lectures on stochastic programming: modeling and theory*, volume 16. SIAM, 2014.
- [125] David B Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *Journal of the ACM (JACM)*, 53(6):978–1012, 2006.
- [126] José A Soto. Matroid secretary problem in the random-assignment model. *SIAM Journal on Computing*, 42(1):178–211, 2013.
- [127] Aravind Srinivasan. Approximation algorithms for stochastic and risk-averse optimization. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 1305–1313, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
- [128] Chaitanya Swamy. Risk-averse stochastic optimization: probabilistically-constrained models and algorithms for black-box distributions. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1627–1646. SIAM, 2011.
- [129] Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, 2006.
- [130] Chaitanya Swamy and David B Shmoys. Sampling-based approximation algorithms for multistage stochastic optimization. *SIAM Journal on Computing*, 41(4):975–1004, 2012.
- [131] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
- [132] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton Univ. Press, 2nd edition, 1947.
- [133] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.
- [134] Yuan Zhou, Xi Chen, and Jian Li. Optimal pac multiple arm identification with applications to crowdsourcing. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 217–225, 2014.