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 xed 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].

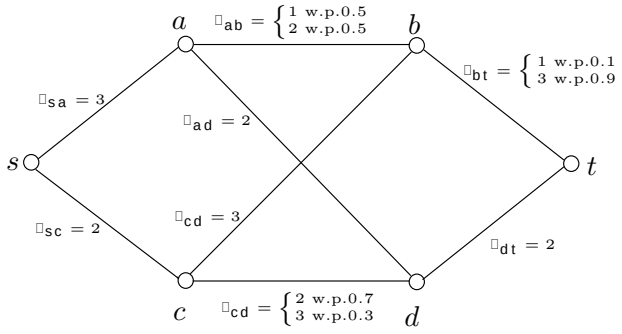
Recent years have witnessed a surge of interest in solving combinatorial optimization problems under uncertainty from the theoretical computer science community. This is in part due to the dramatic increase in the number of application domains, such as information extraction systems, data integration 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 randomization [1]). 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.

[1] We 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.



Realizations	ℓ_{ab}	ℓ_{cd}	ℓ_{bt}	Prob
r_1	1	2	1	.035
r_2	1	2	3	.315
r_3	1	3	1	.015
r_4	1	3	3	.135
r_5	2	2	1	.035
r_6	2	2	3	.315
r_7	2	3	1	.015
r_8	2	3	3	.135

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*

[2] For 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.

[3] Another 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 xed deadline and we want to gain as much pro t as possible before the deadline. The precise processing time and pro t 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 identi ed and solved optimally (see e.g., [42, 43, 9, 8]). The problem has a very di erent nature from the xed 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 xed set stochastic shortest path problem, we make our nal 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 $\mathcal{S}^{\mathrm{I}} \subseteq \mathcal{F}$ at the first stage. In the second stage, the actual demand set U' ($U' \subseteq U$) is revealed. However, our first stage decision $\mathcal{S}^{\mathrm{I}}$ may not be able to cover U', and we can select some extra subsets $\mathcal{S}^{\mathrm{II}} \subseteq \mathcal{F}$ 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* in ation 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 receives 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.

[4] We 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.

[5] It is easy to see that if ≤ 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 at 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

[6] We 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 $(1-\epsilon)$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 $O(n^{1=\ ^2})$ 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 $\frac{1}{}$. 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 $A \in \mathcal{I}$;

Item	Size Distribution(capacity = 1)
1	0.2(prob 0.5) 0.6 (prob 0.5)
2	0.8(prob 1)
3	0.4 (prob 0.5) 0.9 (prob 0.5)

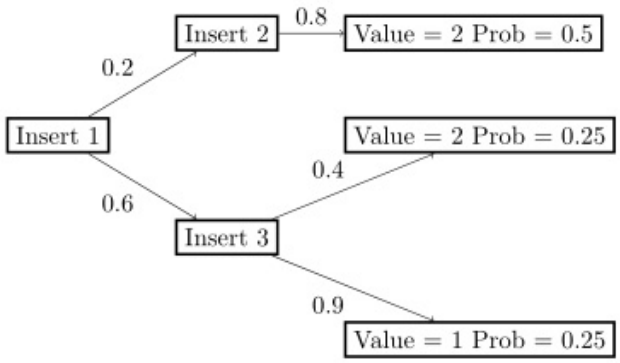


Figure 2: A decision tree corresponding to an adaptive policy.

3. 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 ith 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 $\mathcal{P} : 2^{\mathcal{I}} \times R^+ \to \mathcal{I}$ 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

$$\mu_i = \mathbb{E}[\min\{s_1, 1\}].$$

For a set of item, we de ne $size(S) = \sum_{i \in S} s_i, \mu(S) = \sum_{i \in S} \mu_i$ 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 de nition 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 rst t items chosen by $\mathcal{P}$. We de ne $X_t = \sum_{i \in S_t} (\tilde{s}_i - \mu_i)$. 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 speci ed later. Assume $\mathcal{L} = \{1, 2, 3, \ldots\}$ such that $\frac{v_1}{1} \geq \frac{v_2}{2} \geq \ldots$. Denote by $M_k = \sum_{i=1}^{k} \mu_i$. Let n^* be the maximum number such that $M_{n^*} \leq 1$. We de ne 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

```
Calculate m_1, m_G;
if m_1 ≥ m_G then
    Insert the item which yields m_1 = v_i Pr[s_i ≤ 1];
else
    for i = 1 → n* do
        Insert item i;
        If the knapsack over ows, break;
```

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

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* GRD $\geq m_1$ *and* 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

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

By Markov inequality, $\Pr[\sum_{i=1}^{k} \$_i > 1] \leq \sum_{i=1}^{k} \mu_i = M_k$, for any $k \leq n^*$. Hence, 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 (c) as follows:

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

We can see (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* $\text{OPT}_{\mathcal{L}} \leq$ (2).

Proof. (sketch) Let us x an arbitrary adaptive policy. We should interpret x_i as the probability that item i is successful 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 (2). ∎

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

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

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

Proof. We prove the lemma by picture. m_G corresponds to the area under the staircase. (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 (1), and the base of each triangle is $\mu_i \leq \sigma$. Therefore $m_G \geq \frac{1-}{2}$ (1). The second inequality is due to the concavity of and the last follows from Lemma 3.4. ∎

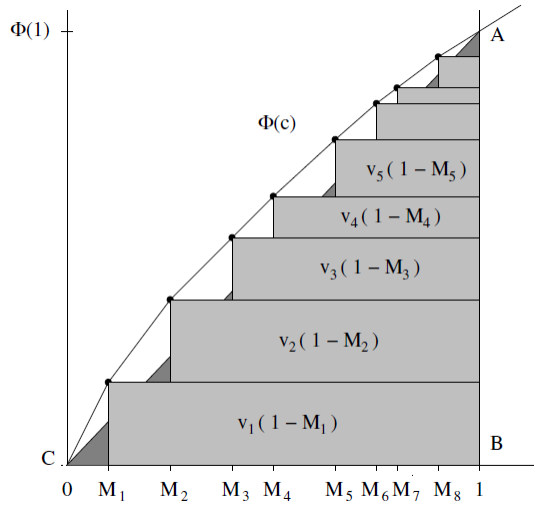


Figure 3: The fractional knapsack solution (c). The light shaded area corresponds to m_G. As (c) is concave, the curve (c) is above the segment AC.

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 $\mathbb{E}[val(H)] \leq \frac{2}{}m_1$.*

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

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

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

Theorem 3.7. *For $\sigma = 1/3$, $\mathrm{OPT} \leq 12\mathrm{GRD}$. 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, $\mathbb{E}[val(L)] \leq \mathrm{OPT}_{\mathcal{L}} \leq \frac{4}{1-}m_G$, and by Lemma 3.6, $\mathbb{E}[val(H)] \leq \frac{3}{}m_1$. Combining with Lemma 3.3, we obtain that

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

■

3.3 Historical Notes, Variants, and Generalizations

The adaptive stochastic knapsack was rst studied by Dean, Goemans and Vondrak [16]. The above greedy algorithm and the analysis are essentially due to them. With some extra e ort, 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 di erence. 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 su cient) 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 Bhalgat, 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 obtained a *bi-criterion PTAS*: their policy can achieve an expected value of $(1 - \epsilon)$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, Bhalgat [19] obtained a $(2 + \epsilon)$-approximation, which is currently the best known approximation factor for the adaptive stochastic knapsack problem. Their approach is further simpli ed 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, base 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 di erent LP which corresponds to a dynamic program, and obtained a polynomial time algorithm that nds 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 a rmative 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 de ned 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 xed 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 $((\log \log B)^{1=2})$, 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 signi cant extension of this technique, called the contention resolution scheme, which can be used to handle several other combinatorial constraints as well.

4.1 Problem De nition

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 brie y describe. The problem and some of its variants also nd 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 pro t, 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.

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

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

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

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

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

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

[9] The 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 $\pi : E \to E$ 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

```
Pick a permutation π on edges E uniformly at random;
for For each edge e in the ordering π do
    If e = (u, v) is not safe then do not probe it;
    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 $\frac{1}{2}$, and

(b) the probability that vertex u is matched is at most $\frac{1}{2}$.

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.

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

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 at 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,

[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

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

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_es 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 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 di cult and can be found in [108].

If 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 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 $\frac{3}{11}$.

Enumerating Heavy Elements: We enumerate all possible set of heavy edges with size at most $3/\epsilon^{11}$. There are at most $n^{3=\ 11}$ 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 rst 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 $\widetilde{\mu}_e$ to denote the size after discretization and $\widetilde{\pi}_e$ its distribution. The discretization consists of following two steps.

1. Small size region: In the small size region, $\widetilde{\mu}_e$ follows a Bernoulli distribution, taking only values 0 and ϵ^4. The probability values Pr[$\widetilde{\mu}_e = 0$] and Pr[$\widetilde{\mu}_e = \epsilon^4$] are set such that $\mathbb{E}[\widetilde{\mu}_e \mid \mu_e \leq \epsilon^4] = \mathbb{E}[\mu_e \mid \mu_e \leq \epsilon^4]$.

2. Large size region: If μ_e realizes to a large size, we simply discretize it as follows: Let $\widetilde{\mu}_e = \lfloor \frac{e}{5} \rfloor \epsilon^5$ (i.e., we round a large size down to a multiple of ϵ^5).

   We denote the set of the discretized sizes by $\mathcal{S} = \{s_0, s_1, \ldots, s_{z-1}\}$ where $s_0 = 0, s_1 = \epsilon^5, s_2 = 2\epsilon^5, s_3 = 3\epsilon^5, \ldots, s_{z-1}$. Note that $s_1 = \epsilon^5, \ldots, s_{1=\ -1} = \epsilon^4 - \epsilon^5$ 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 nishes the description of the discretization.

It is not di cult 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.

[13] As $\to 0$, the discretizated 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 $\mathbb{E}[\mu(S)] \le 3/\epsilon$. We have that*

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

8.2 Poisson Approximation

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

$$\mathrm{Sg}(e) = \big(\overline{\pi}_e(s_1), \overline{\pi}_e(s_2), \overline{\pi}_e(s_3), \ldots, \overline{\pi}_e(s_{z-1})\big),$$

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

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

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

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

Obviously, if $(X, Y) = 0$, 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 $\mathrm{Sg}(S_1) = \mathrm{Sg}(S_2)$ and $\mathbb{E}\left[\widetilde{X}(S_1)\right] \le 3/\epsilon, \mathbb{E}\left[\widetilde{X}(S_2)\right] \le 3/\epsilon$. Then, the total variation distance between $X(S_1)$ and $X(S_2)$ satisfies*

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

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 $V = (V_1, \ldots, V_K)$. 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:

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

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

Lemma 8.5. [105] *Let $X_1, X_2, \ldots$ be independent random variables taking integer values in $\{0, 1, ..., K\}$, let $X = \sum X_i$. Let $\pi_i = \Pr[X_i \neq 0]$ and $V = (V_1, \ldots, V_K)$ 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:*

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

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

$$\Big(\overline{X}(S_1), \widetilde{X}(S_1)\Big) \leq \epsilon \qquad \text{and} \qquad \Big(\overline{X}(S_2), \widetilde{X}(S_2)\Big) \leq \epsilon.$$

First, for any S such that $\mathbb{E}\left[\widetilde{X}(S)\right] \leq 3/\epsilon$, we can see that $\sum_{b \in S} \Pr\left[\widetilde{\mu}_e \neq 0\right] \leq \mathbb{E}\left[\widetilde{\mu}(S)\right]/\epsilon^4 \leq 3/\epsilon^5$. If we apply Lemma 8.5 to both $\overline{X}(S_1)$ and $\overline{X}(S_2)$, 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

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

This nishes the proof of the lemma. ■

8.3 Algorithm for Fixed Set Problems

Now, we present our approximation algorithm for the xed 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 $\mathbb{E}\big[\widetilde{X}(H)\big] < 3/\epsilon$;
3. for *each such* H do
4. Enumerate all possible signatures Sg;
5. for *each such* Sg do
6. Try to nd an s-t path $S = H \cup L$, such that $\mathrm{Sg}(L) = \mathrm{Sg}$;
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 nd a set L with the signature exact equal to Sg. This can be done by standard dynamic programming. [14] Since 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 $n^{\mathrm{poly}(1=\epsilon)}$. Thus the pseudopolynomial time algorithm actually runs in $\mathrm{poly}(n, n^{\mathrm{poly}(1=\epsilon)})$ time, which is a

[14] We 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 su ces for our purpose. There are a polynomial number of combinations of heavy edges. So it is not hard to see the algorithm runs in $n^{\mathrm{poly}(1=\ )}$ 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 $\widetilde{\mu}(L)$ ($\widetilde{\mu}(L)$ resp.). By Lemma 8.4, $\widetilde{\mu}(L)$ is close to the $\widetilde{\mu}(L^*)$. 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 rst sight where to apply the Poisson approximation technique, which can only handle the sum of a xed set of random variables.

We need an important notion, called *block-adaptive policies*, introduced by Bhalgat 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 pro t 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 jobs [15] and precedence constraints. Hence, it su ces to consider only block-adaptive policies. Since there are only constant number of nodes in the decision tree, we can enumerate all di erent 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 approximation the distribution of the sum of random variables in the block. More concretely, after xing 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 pro t).

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

[15] In 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 $\mathcal{U} : \mathbb{R}^+ \to \mathbb{R}^+$. 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 a 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 $\left|\mathcal{U}(x) - \mathcal{C}(x)\right| \leq \epsilon, \ \ \forall x \geq 0$. 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, Bhalgat 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

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

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 discretizaton 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.

[16] If 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 secretrary 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.

[17] There 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 $\left(\sum_{i=2}^{n} \Delta_i^{-2} \ln \delta^{-1}\right)$ samples in expectation for any instance, where Δ_i is the difference between the mean of the best arm and that of the ith arm. Chen and Li [34] obtained the current best upper bound

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

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 optimal [18] (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 $\Theta(n \ln \delta^{-1}/\epsilon^2)$ [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).

[18] Instance 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 xed 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 problem [19] for the locational model in a nite 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 xed dimensional Euclidean space and provided a PTAS. Recently, Huang et al. [84] obtained a PTAS for the more general j- at center problem (i.e., the center is a j- at, i.e., a j-dimensional a ne subspace) for stochastic point, using an extension of the powerful geometric notation *ϵ-kernel coreset*, 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.

[19] The 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. Babaio , 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, Julian 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 Vondrak. Approximating the stochastic knapsack problem: the bene t 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 Jose 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 + ϵ)-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] Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1{122, 2012.

[24] Sebastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identi cations in multi-armed bandits. *arXiv preprint arXiv:1205.3181*, 2012.

[25] Niv Buchbinder, Joseph Se 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] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvo , and Laura Sanita. 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 Gabor 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 Pal. 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 Vondrak, 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 identi cation. *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 Vondrak. 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 Vondrak. Approximating the stochastic knapsack problem: The bene t 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 Cli 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 (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 Konemann, Stefano Leonardi, and Guido Schafer. 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 identi cation: A uni ed approach to xed budget and xed con dence. In *Advances in Neural Information Processing Systems*, pages 3212{3220, 2012.

[60] Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sebastien Bubeck. Multi-bandit best arm identi cation. 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 in ation. 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 in ation. 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 Pal, 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 Pal, 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, Je 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 Minko , and Vahab S Mirrokni. On the costs and bene ts 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 Sebastien 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. E cient 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 Karget and Maria Minko . 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 Cappe, and Aurelien Garivier. On the complexity of best arm identi cation 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 Eva 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 nite value. *Advances in Prob*, 4:197{266, 1978.

[104] Oded Lachish. O (log log 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 combinatirial 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 o ine 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] Jose 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 identi cation with applications to crowdsourcing. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 217{225, 2014.