

Stochastic Extreme Value Optimization

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Computer Science and Technology

by

Liu Yu

Thesis Supervisor: Assistant Professor Li Jian

January 2017

Stochastic Extreme Value Optimization

by

Liu Yu

Submitted to the Institute for Interdisciplinary Information Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

at

TSINGHUA UNIVERSITY

January 2017

© TSINGHUA UNIVERSITY 2017. All rights reserved.

Author
Institute for Interdisciplinary Information Sciences
Oct 28, 2016

Certified by
Li Jian
Assistant Professor
Thesis Supervisor

Abstract

In this thesis, we study several stochastic optimization problems in which we want to optimize the expectation of the extreme values, including MAXIMUM-ELEMENT, minimization version of MAXIMUM-ELEMENT and MINIMUM-ELEMENT. Formally, we obtain the following results:

1. We first study MAXIMUM-ELEMENT. It takes a set of random variables $\{X_i\}_{i=1}^n$ (with known distributions) and a collection of feasible sets (constraint) $\mathcal{F} \subset 2^{[n]}$ as inputs, and outputs a feasible set $S \in \mathcal{F}$ which maximizes the value of $\mathbb{E}[\max_{i \in S} X_i]$. In the minimization version of MAXIMUM-ELEMENT, we try to minimize the value of $\mathbb{E}[\max_{i \in S} X_i]$.

We obtain a constant approximation for MAXIMUM-ELEMENT if there is a constant approximation for the deterministic problem MAX-SUM (find a feasible set $S \in \mathcal{F}$ which maximizes the value of $\sum_{i \in S} v_i$). Compared with previous results, our approximation works for many combinatorial constraints. Furthermore, we obtain the first PTAS for MAXIMUM-ELEMENT if there is a pseudopolynomial time algorithm for the EXACT-SUM problem. The best previous approximation ratio is $1 - 1/e$ [6], based on submodular maximization.

2. We can obtain the same constant approximation and PTAS for the minimization version of MAXIMUM-ELEMENT, by a slight modification of the above algorithm.
3. We also study the MINIMUM-ELEMENT problem, which takes a set of random variables $\{X_i\}_{i=1}^n$ with costs $\{c_i\}_{i=1}^n$ and a budget C as inputs, and outputs a set S with total cost at most C which minimizes the value of $\mathbb{E}[\min_{i \in S} X_i]$. If all variables are supported on $\{0, 1, \dots, m-1\}$, we obtain a bi-criteria algorithm which outputs a $(1+\varepsilon)$ -approximation with cost at most $O(\log \log \log m)C$, improving the best previous result which incurs a cost at most $O(\log \log m)C$ [10].

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Related Work	2
1.3	Our Contributions	3
1.3.1	MAXIMUM-ELEMENT	5
1.3.2	MINIMUM-ELEMENT	6
1.4	Structure of this Thesis	7
2	Preliminaries	9
2.1	Terminologies	9
3	MAXIMUM-ELEMENT	13
3.1	Problem Formulation	13
3.2	Hardness	14
3.3	A Constant Approximation	15
3.3.1	Special Case: Bernoulli Distributions	16
3.3.2	General Cases	18
3.4	PTAS	22
3.4.1	Discretization	22
3.4.2	Signatures	25
3.4.3	Enumerating Signatures	27
3.5	Minimization version of MAXIMUM-ELEMENT	28

3.6	MAX_K	31
4	MINIMUM-ELEMENT	35
4.1	Problem Formulation	35
4.2	Hardness	35
4.3	CIP Approach	36
4.4	Our CIP Approach	37
5	Conclusion	45
	Bibliography	47
	Curriculum Vitae	52

Chapter 1

Introduction

1.1 Motivation

Recent years data sets grow rapidly. They are increasingly collected by mobile devices, cameras, remote sensors, software logs and other cheap and numerous information-sensing devices. Along with the increasing of the size of data collected, uncertainty arises in many applications, including social networks, sensor networks, databases, social science economics and so on. Many applications require supports of methods for dealing with uncertain data. Here we give some examples to show where the uncertainty comes and how we model them.

- **Bayesian Optimal Mechanism Design [19]:** In economics, mechanism design (in particular auctions) is an important research area. A good mechanism should allocate the items and decides the payment based on the private value of the items of each bidder. However no one knows the exact value except the bidder herself. Hence, we model her value as a random variable following some prior distribution (probably computed from historical data of the bidder), and design mechanisms according to the distributions.
- **Sensor Network:** Sensor networks spread everywhere and produce large amount of data. Due to physical noise or detection precision, much uncertain data is produced at the same time. In such case, uncertain sensor readings are usually modeled by

probabilistic models. For example, Deshpande et al. [8] modeled data as Gaussian distributions.

Stochastic optimization is a method for tackling uncertainty by modeling uncertainty as probability distributions of inputs. The field was initiated by Dantzig [7]. In the following decades, it has been studied extensively, and attracts more and more attentions from computer science community in recent years. In my dissertation research, I focus on one class of important problems in this domain: stochastic extreme value optimization.

Stochastic extreme value optimization: In a large variety of practical problems, people concern the extreme value (i.e., the maximum value, the minimum value). Here we list some examples.

- In a first price auction, the price of the item is determined by the highest bid.
- Consider a scenario where one agent wants to produce some object. He delivers the jobs of producing components to different factories. The factory with the latest complete time determines the complete time of the whole production.

In such problems, people want to optimize an extreme value as it affects results significantly. When involving uncertain data, we call such problems “stochastic extreme value optimization”.

1.2 Related Work

In the past decades the stochastic combinatorial optimization has been widely and deeply studied. Many combinatorial optimization problems themselves are NP-hard. The problem usually looks harder involving uncertainty and people proved some of them are actually harder. For example, a variant of adaptive stochastic knapsack is PSPACE-hard [3]. While an optimal solution is hard to find, people make efforts to find approximate solutions efficiently. People have developed many techniques to deal with approximating problems in the

field. More and more problems are studied in different models: Adaptive stochastic knapsack [3, 17], stochastic matching [1, 2], stochastic set cover [22, 13], stochastic bin packing [17] and so on. We refer interested reader to [16] for a comprehensive survey.

In some stochastic optimization problems, it is possible to make observation on inputs. Upon one observation of one input item, we can find out the exact value (which is a sample from the given distribution). Observations need extra cost. Hence we should decide which inputs to observe to maximize the improvement using as little cost as possible. Goel et al. [10] modeled such situations and defined a class of stochastic optimization problems called “model-driven optimization”, which generalizes the stochastic extreme value optimization with knapsack constraint.

1.3 Our Contributions

First we define two stochastic extreme value optimization problems MAXIMUM-ELEMENT and MINIMUM-ELEMENT formally here.

Problem 1.3.1 (MAXIMUM-ELEMENT). *Inputs: a set of independent non-negative random variables $\{X_i\}_{i=1}^n$ and a collection of feasible solutions (a combinatorial constraint) $\mathcal{F} \subset 2^{[n]}$; Outputs: a set $S \in \mathcal{F}$, which maximizes the value $\mathbb{E}[\max_{i \in S} X_i]$.*

Note that \mathcal{F} may contain exponential number of feasible sets and is represented implicitly in the input. For example, each random variable is associated with an edge of a graph and \mathcal{F} is the set of all spanning trees of the graph. Distributions of random variables can be inputted in two ways: lists of supports and corresponding probabilities, which works for discrete random variables; and oracles returning samples following the corresponding distributions, which works for both discrete and continuous random variables. In this thesis, we only consider discrete random variables, leaving the case of continuous random variables as future work.

Problem 1.3.2 (MINIMUM-ELEMENT). *Inputs: a set of independent non-negative random variables $\{X_i\}_{i=1}^n$, costs $\{c_i\}$, and a budget C ;*

Outputs: a set S with total cost at most C , which minimizes the value $\mathbb{E}[\min_{i \in S} X_i]$.

In the minimization version of MAXIMUM-ELEMENT, we want to minimize the value $\mathbb{E}[\max_{i \in S} X_i]$.

Relationship between MAXIMUM-ELEMENT and MINIMUM-ELEMENT: Given a set of random variables $\{X_i\}$, we can construct a set of random variables $\{Y_i\}$ with $Y_i = M - X_i$, where M is large enough to make sure all Y_i 's are non-negative. Therefore $\max_{i \in S} X_i = M - \min_{i \in S} Y_i$, and solving MAXIMUM-ELEMENT over $\{X_i\}$ is equivalent to solving MINIMUM-ELEMENT over $\{Y_i\}$. When finding the exact solution, these two problems are both NP-hard. However, they are quite different in approximation hardness.

We also define three related problems for stating our results.

Problem 1.3.3 (MAX-SUM). *Inputs: a set of non-negative real numbers $\{v_i\}_{i=1}^n$ and a constraint $\mathcal{F} \subset 2^{[n]}$;*

Outputs: a set $S \in \mathcal{F}$, which maximizes the value $\sum_{i \in S} v_i$.

Problem 1.3.4 (MIN-SUM). *Inputs: a set of non-negative real numbers $\{v_i\}_{i=1}^n$ and a constraint $\mathcal{F} \subset 2^{[n]}$;*

Outputs: a set $S \in \mathcal{F}$, which minimizes the value $\sum_{i \in S} v_i$.

MAX-SUM and MIN-SUM generalizes many classical problems, such as maximum matching, max cut, minimum spanning tree, shortest path.

Problem 1.3.5 (EXACT-SUM). *Inputs: a set of integers $\{v_i\}$, a target integer T and a constraint $\mathcal{F} \subset 2^{[n]}$;*

Outputs: a set $S \in \mathcal{F}$ with $\sum_{i \in S} v_i = T$.

MAXIMUM-ELEMENT and MINIMUM-ELEMENT are first introduced by Goel et al. in [10] as special cases of their Model-driven Optimization problem, defined as follows:

Problem 1.3.6. *We are given the distributions of non-negative independent random variables $\{X_i\}_1^n$. These variables are observable: we can find the exact value of X_i by cost c_i .*

Given an objective function $f : 2^{\{X_i\}_1^n} \rightarrow \mathcal{R}$ and budget C , the goal is to choose a set of variables S with cost at most C to observe and optimize the expected value of $f(S)$. The function is evaluated after observations and the expectation is over all possible outcome of observations.

1.3.1 MAXIMUM-ELEMENT

As this problem is NP-complete, people studied approximation algorithms. Goel et al. [9] studied two variants of knapsack problem, which can generalize MAXIMUM-ELEMENT, and come up with a $\frac{1}{4}(1 - \frac{1}{e})$ -approximation algorithm respecting the knapsack constraint $\{S \mid \sum_{i \in S} w_i \leq W\}$. Later, Guha et al. [12] considered the Lagrangian version of MAXIMUM-ELEMENT, where the cost is part of objective value. They show that the Lagrangian version can be solved optimal. Chen et al. [6] studied an online version with cardinality constraint and obtained an online algorithm with regret bounded. They also obtained a $(1 - \frac{1}{e})$ -approximation offline algorithm for cardinality constraint $\{S \mid |S| = K\}$.

Our results on MAXIMUM-ELEMENT: Using the truncation idea in [14, 11], we obtain a constant approximation. Further we make use of the discretization technique used in [15, 17] and obtain the first PTAS of this problem. We obtain the same constant approximation and PTAS for the minimization version of MAXIMUM-ELEMENT by a slight modification of the algorithms for MAXIMUM-ELEMENT.

- Theorem 1.3.7.** *1. If there is a α -approximation algorithm for MAX-SUM, there is a $\frac{\alpha}{(1+\alpha)(1+\varepsilon)}$ -approximation algorithm for MAXIMUM-ELEMENT for any constant $\varepsilon < 0.5$.*
- 2. If there is a α -approximation algorithm for MIN-SUM, there is a $\frac{1+\alpha}{1+\varepsilon}$ -approximation algorithm for minimization version of MAXIMUM-ELEMENT for any constant $\varepsilon < 0.5$.*
- 3. If there is a pseudopolynomial time algorithm for EXACT-SUM, there is a PTAS for both MAXIMUM-ELEMENT and the minimization version of MAXIMUM-ELEMENT.*

In previous work, all optimization bases on that $f(S) = \mathbb{E}[\max_{i \in S} X_i]$ is a submodular set function. Under constraints such as cardinality, knapsack, matroid, a greedy algorithm for submodular function maximization is a $(1 - \frac{1}{e})$ -approximation[5]. But our algorithm for MAXIMUM-ELEMENT only provides a $(\frac{1}{2} - \varepsilon)$ -approximation.

However, as we only require constant approximation for MAX-SUM and MIN-SUM, our constant approximation algorithms work for more constraints. For example, consider the cut constraint, where each variable is related to a vertex in a graph and all cuts of the graph are feasible sets. MAX-CUT can be approximated with ratio 0.878 [23] and MIN-CUT can be solved optimal [21]. Hence Theorem 1.3.7 provides constant approximation algorithms for both MAXIMUM-ELEMENT and the minimization version, while previous work can not handle such constraint.

The best result in previous work is a constant approximation [6]. Our result is the first PTAS for this problem. We require EXACT-SUM to have a pseudopolynomial time algorithm. Here we give two examples where dynamic programming admits a pseudopolynomial time algorithm.

- Knapsack: Each variable has a weight w_i and all sets with total weight at most W are feasible sets.
- Sub-Tree: Each variable is associated to a node in a rooted tree. All sub-trees are feasible sets.

1.3.2 MINIMUM-ELEMENT

Goel et al. [10, 9] studied MINIMUM-ELEMENT deeply. They proved the hardness of MINIMUM-ELEMENT and proposed the Covering Integer Programs (CIP) approach for approximating the cost in MINIMUM-ELEMENT.

Theorem 1.3.8. [9] *It is NP-hard to obtain any polynomial approximation for MINIMUM-ELEMENT without violating the budget.*

Theorem 1.3.9. [10] *To achieve $(1 + \varepsilon)$ -approximation on objective value, if the inputs are*

1. *distributions with upper bound M , there is a solution with cost $O(\log \log \frac{M}{V})C$ where V is a bound to OPT.*
2. *discrete distributions with number of supports m , there is a solution with cost at most $O(\log m)C$. Moreover, if the supports of variables is $\{0, 1, \dots, m - 1\}$, the cost is at most $O(\log \log m)C$.*
3. *log-concave distributions and uniform distributions, there is a solution with cost at most $O(1)C$.*

Our results on MINIMUM-ELEMENT: Following the previous work[10], we design an algorithm with better CIPs. We combine CIP approach and the discretization technique. We improve the cost to $O(\log \log \log m)C$ from $O(\log \log m)C$ in the case where the supports of variables is $\{0, 1, \dots, m - 1\}$.

Theorem 1.3.10. *When $\{X_i\}$ is supported on $\{0, 1, \dots, m - 1\}$, there is a polynomial time algorithm providing a $(1 + \varepsilon)$ -approximation with cost at most $O(\log \log \log m)C$.*

1.4 Structure of this Thesis

The rest of this thesis is organized as follows. Chapter 2 provides some basic mathematics and computer science knowledge and terminologies that will be used in this thesis. We will show our results of MAXIMUM-ELEMENT in Chapter 3, part of results in this chapter appeared in [6]. And we show the results of MINIMUM-ELEMENT in Chapter 4. Finally, we conclude this thesis in Chapter 5.

Chapter 2

Preliminaries

2.1 Terminologies

Here we review some related terminologies which will be used in later parts of this thesis.

Distributions and Random Variables: A random variable X follows a probability distribution \mathcal{D} is denoted by $X \sim \mathcal{D}$. The support of one distribution $\text{Supp}(\mathcal{D})$ is the smallest closed set whose complement has probability zero. If $\text{Supp}(\mathcal{D})$ is a set of discrete values, we call this distribution discrete distribution. If all values in $\text{Supp}(\mathcal{D})$ is non-negative, we call it non-negative distribution. Here are some examples of discrete distributions.

- Bernoulli Distribution: A Bernoulli distribution $B(a, p)$ with parameters a, p takes value a with probability p and value 0 with probability $1 - p$. For simplicity, we also use $B(a, p)$ to denote the Bernoulli variable which follows the Bernoulli distribution $B(a, p)$ in this thesis.
- Poisson Distribution: A Poisson Distribution $Poi(\lambda)$ with parameter λ takes value on non-negative integers, and $\Pr[Poi(\lambda) = k] = \frac{\lambda^k}{k!} e^{-\lambda}$.

For two distributions P and Q with $\text{Supp}(P) = \text{Supp}(Q) = S$, the distance $L_1(P, Q)$ is defined by $\sum_{v \in S} |\Pr[P = v] - \Pr[Q = v]|$.

Concentration Bounds: The following concentration bounds will be useful in the thesis (see e.g., [18])

- Markov inequality. Let X be a random variable and $\mathbb{E}[X]$ be its expectation. Then for any positive α ,

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

- Chernoff Bound. Let X_1, \dots, X_n be independent random variables taking values in $\{0, 1\}$. Let X be their sum and $\mu = \mathbb{E}[X]$. Then

$$\Pr[X \geq (1 + \delta)\mu] \leq e^{-\frac{\delta^2\mu}{3}}$$

and

$$\Pr[X \leq (1 - \delta)\mu] \leq e^{-\frac{\delta^2\mu}{2}}$$

for any $0 < \delta < 1$.

Matroid[4]: Matroid is an important structure widely considered in computer science.

Definition 2.1.1 (Matroid). *A finite matroid \mathcal{M} is a pair $(\mathcal{U}, \mathcal{I})$, where \mathcal{U} is the universe set (ground set) and \mathcal{I} is a family of subset of \mathcal{U} (called independent sets), with the following properties:*

- *The empty set is independent;*
- *Any subset of independent set is independent;*
- *If A and B are independent sets and A has more elements than B , there exists an element in A and adding it to B makes a larger independent set than B .*

Matroid generalizes many combinatorial objects. Here are some examples for matroids.

- Uniform Matroid. Let all subsets of the ground set with size at most k be independent sets. It is easy to check the three properties.

- Forest Matroid. In a graph, let the set of edges be the ground set and take all forests in the graph be independent sets. It forms a forest matroid of a graph.

Submodular Functions: Submodular functions generalizes a number of objective functions of well-known combinatorial optimization problems (see e.g., [20]).

Definition 2.1.2 (Submodular Function). *A function $f : 2^{[n]} \rightarrow \mathcal{R}$ is called submodular, if and only if*

$$\forall S, T \subset [n], f(S) + f(T) \geq f(S \cap T) + f(S \cup T).$$

It is called monotone if $\forall T \subset S, f(T) \leq f(S)$ or $\forall S \subset T, f(T) \leq f(S)$.

Here are some examples for submodular functions.

- Linear Functions. $f(S) = \sum_{i \in S} w_i$ for some weights $\{w_i\}$. It is easy to check that $f(S) + f(T) = f(S \cap T) + f(S \cup T)$.
- Set Cover. Let $\{S_1, \dots, S_n\}$ be a collection of subsets of some set G . The function $f(T) = |\cup_{i \in T} S_i|$ is submodular.

Approximation Ratio: To evaluate the performance of an approximation algorithm, approximation ratio is introduced. For a minimization problem, we say an algorithm achieves approximation ratio of $\alpha \geq 1$ if

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

where SOL denotes the cost of the solution found by the algorithm, OPT denotes the optimal cost and the expectation is over all randomness of the problem inputs and the algorithm itself. Similarly, for maximization problem, we say an algorithm achieves approximation ratio of $\alpha \leq 1$ if

$$\mathbb{E}[\text{SOL}] \geq \alpha \text{OPT}.$$

A polynomial time approximation scheme (PTAS) is an algorithm which takes an instance of a problem and a parameter ε and achieves approximation ratio $1 + \varepsilon$ or $1 - \varepsilon$, and the

running time is polynomial in the size of input, for fixed ε . We say a PTAS is a fully polynomial time approximation scheme (FPTAS) if the running time is also polynomial in $\frac{1}{\varepsilon}$. A pseudopolynomial time algorithm is an algorithm with running time being polynomial in the numeric value of the inputs, which is exponential to the size of input.

Adaptivity: When designing algorithm for stochastic optimization problems, according to when the exact value of uncertain data is revealed, there are two kinds of models.

1. Adaptive: Uncertainty is revealed immediately when one decision is made, and the next step of the algorithm can dependent on this information. In other words, once the algorithm take some operation o_i , it can observe some uncertain data and get its exact value v_i , then make next decision according to o_i, v_i and all previous information.
2. Non-adaptive: Algorithms do not base their decisions on any revealed information. All decisions are made in advance and off-line. In other words, it decides operations o_1, o_2, \dots and performs operations in such order, no matter what exact value of uncertain data is.

Usually adaptive algorithms perform better than non-adaptive ones. In this thesis, we only consider the non-adaptive model.

Chapter 3

MAXIMUM-ELEMENT

We formulate and solve the MAXIMUM-ELEMENT problem in Chapter 3. In this chapter, we present some non-adaptive results. First, we present our constant approximation for MAXIMUM-ELEMENT. After that we show our PTAS. Finally, we try to extend our constant approximation result to the Top-K optimization and leave a conjecture for future work.

3.1 Problem Formulation

We give the formulation of MAXIMUM-ELEMENT here.

Problem 3.1.1 (MAXIMUM-ELEMENT). *Inputs: a set of independent non-negative random variables $\{X_i\}_{i=1}^n$ with known distributions and a collection of feasible solutions (constraint) $\mathcal{F} \subset 2^{[n]}$;*

Outputs: a set $S \in \mathcal{F}$, which maximizes the value of $\mathbb{E}[\max_{i \in S} X_i]$.

The minimization version of MAXIMUM-ELEMENT minimizes the value $\mathbb{E}[\max_{i \in S} X_i]$. Here X_i follows a discrete distribution D_i with support $\{a_{i,j}\}_{j=1}^{s_i}$ ($s_i = \text{poly}(n)$). Let $p_{i,j} = \Pr[X_i = a_{i,j}]$. The inputs of MAXIMUM-ELEMENT are $\{a_{i,j}\}$, $\{p_{i,j}\}$ and \mathcal{F} . Note that \mathcal{F} may contain exponential number of feasible sets and is represented implicitly in the input. For example, each random variable is associated with an edge of a graph and \mathcal{F} is the set of all spanning trees of the graph. This is our main theorem for MAXIMUM-ELEMENT.

- Theorem 3.1.2.** 1. If there is a α -approximation algorithm for MAX-SUM, there is a $\frac{\alpha}{(1+\alpha)(1+\varepsilon)}$ -approximation algorithm for MAXIMUM-ELEMENT for any constant $\varepsilon < 0.5$.
2. If there is a α -approximation algorithm for MIN-SUM, there is a $\frac{1+\alpha}{1+\varepsilon}$ -approximation algorithm for minimization version of MAXIMUM-ELEMENT for any constant $\varepsilon < 0.5$.
3. If there is a pseudopolynomial time algorithm for EXACT-SUM, there is a PTAS for both MAXIMUM-ELEMENT and the minimization version of MAXIMUM-ELEMENT.

3.2 Hardness

The MAXIMUM-ELEMENT problem is NP-Complete. As we said in the introduction, it can be implied by the hardness of MINIMUM-ELEMENT problem. Here we provide a separate proof, which is clean and easy to understand.

Theorem 3.2.1. *The MAXIMUM-ELEMENT problem is NP-Complete for the total weight constraint $\mathcal{F} = \{S \mid \sum_{i \in S} w_i \leq 1\}$.*

Proof. To show the NP-Completeness, we define an equivalent yes-or-no problem for MAXIMUM-ELEMENT: given a value B , is there a solution of MAXIMUM-ELEMENT with value at least B ? We first show it is in NP and reduce the knapsack problem to it.

To show MAXIMUM-ELEMENT is in NP, we show that given a set S , the value $\mathbb{E}[\max_{i \in S} X_i]$ can be calculated in polynomial time. Recall that each X_i follows a discrete distribution \mathcal{D}_i with support $\{a_{i,j}\}_{j=1}^{s_i}$ ($s_i = \text{poly}(n)$). Note that $\max_{i \in S} X_i$ can only take value in the set $V(S) = \{a_{i,j} \mid i \in S, j \leq s_i\}$, which is of polynomial size. For any $v \in V(S)$, we have

$$\begin{aligned} \Pr[\max_{i \in S} X_i = v] &= \Pr[\max_{i \in S} X_i \leq v] - \Pr[\max_{i \in S} X_i < v] \\ &= \prod_{i \in S} \Pr[X_i \leq v] - \prod_{i \in S} \Pr[X_i < v]. \end{aligned}$$

The last equation follows from that all random variables are independent. So for any $v \in V(S)$, $\Pr[\max_{i \in S} X_i = v]$ can be calculated in polynomial time and $\mathbb{E}[\max_{i \in S} X_i]$ can be

calculated in polynomial time by

$$\mathbb{E}[\max_{i \in S} X_i] = \sum_{v \in V(S)} v \cdot \Pr[\max_{i \in S} X_i = v].$$

For the reduction, consider an instance of the knapsack problem: the items have weights $\{w_i\}_{i=1}^n$ and values $\{v_i\}_{i=1}^n$, the knapsack has capacity 1 and the target value is V . We construct a set of Bernoulli variables $\{X_i\}$ for MAXIMUM-ELEMENT, where $X_i = B(1, 1 - e^{-v_i})$. X_i has weight w_i and the constraint is $\mathcal{F} = \{S \mid \sum_{i \in S} w_i \leq 1\}$. According to our construction, we have

$$\begin{aligned} \mathbb{E}[\max_{i \in S} X_i] &= \Pr[\max_{i \in S} X_i = 1] \\ &= 1 - \Pr[\max_{i \in S} X_i = 0] \\ &= 1 - \prod_{i \in S} \Pr[X_i = 0] \\ &= 1 - \prod_{i \in S} e^{-v_i} \\ &= 1 - e^{-\sum_{i \in S} v_i} \end{aligned}$$

Let $B = 1 - e^{-V}$. MAXIMUM-ELEMENT has a solution with value at least B is equivalent to that the knapsack problem has a solution with value at least V . It is well-known that knapsack problem is NP-Complete, hence the MAXIMUM-ELEMENT problem is NP-Complete. ■

3.3 A Constant Approximation

As the MAXIMUM-ELEMENT problem is NP-Complete, we consider the problem of designing a constant approximation algorithm. The following theorems provide constant approximation algorithms under some constraints.

Theorem 3.3.1. [6] *For a set of random variables $\{X_i\}_{i=1}^n$, the function $f(S) = \mathbb{E}[\max_{i \in S} X_i]$*

defined over subsets of $[n]$ is monotone and submodular.

Theorem 3.3.2. [5] *There are constant factor approximation algorithms for maximizing monotone submodular functions under the following constraints:*

- $O(1)$ knapsacks $[1 - \frac{1}{e} - \varepsilon]$;
- k matroids $[\frac{0.38}{k}]$;
- l -sparse packing systems $[\Omega(\frac{1}{l})]$;
- the intersection of constant number of above constraints.

Numbers in square brackets are corresponding approximation ratios.

Different from the general approximation for submodular function maximization, we propose a new constant approximation algorithm for MAXIMUM-ELEMENT problem, which is easier to apply and also works under other constraints.

3.3.1 Special Case: Bernoulli Distributions

We begin with a special case: all the random variables follow Bernoulli distributions, i.e., $X_i = B(a_i, p_i)$. Before presenting the algorithm, we first define a function we will use.

Definition 3.3.3 (Truncation Function). [11] *For a random variable X , define the truncation function $\text{Tr}(X, T) = \max\{X - T, 0\}$.*

For a set of random variables $\{X_i\}_{i=1}^n$ and a set $S \subset [n]$, define $\text{Tr}(S, T) = \sum_{i \in S} \text{Tr}(X_i, T)$.

The pseudocode of the algorithm can be found in Algorithm 1, where \mathcal{A} is a polynomial time algorithm solving MAX-SUM with approximation ratio $\alpha \leq 1$. Algorithm 1 provides a constant approximation if α is a constant.

Theorem 3.3.4. *Algorithm 1 returns a set S with approximation ratio $\frac{\alpha}{(1+\alpha)(1+\varepsilon)}$, and it terminates in polynomial time.*

Algorithm 1 Constant Approximation for Bernoulli Case

Require: $(a_1, a_2, \dots, a_n, p_1, p_2, \dots, p_n, \mathcal{F})$

- 1: $T \leftarrow \max a_i$
 - 2: $S \leftarrow \mathcal{A}(\mathbb{E}[\text{Tr}(X_1, T)], \dots, \mathbb{E}[\text{Tr}(X_n, T)], \mathcal{F})$
 - 3: **while** $\mathbb{E}[\text{Tr}(S, T)] \leq T$ **do**
 - 4: $T \leftarrow T/(1 + \varepsilon)$
 - 5: $S \leftarrow \mathcal{A}(\mathbb{E}[\text{Tr}(X_1, T)], \dots, \mathbb{E}[\text{Tr}(X_n, T)], \mathcal{F})$
 - 6: **end while**
 - 7: Return S
-

Proof. Before prove Theorem 3.3.4, we provide a useful lemma, which is in a similar form of Lemma 3.3 in [14]. Compared with it, we take a stronger condition and provide a better result.

Lemma 3.3.5. *Given a fixed number T , if $\mathbb{E}[\text{Tr}(S, T)] \geq T$,*

$$\mathbb{E}[\max_{i \in S} X_i] \geq T. \quad (3.1)$$

Proof of Lemma 3.3.5. Assume $S = [N]$ and $a_1 \geq a_2 \geq \dots \geq a_N > T$. We prove that for any given $\{p_i\}$, (3.1) holds.

When $N = 1$, $(a_1 - T)p_1 = \mathbb{E}[\text{Tr}(S, T)] \geq T \Rightarrow \mathbb{E}[\max_{i \in S} X_i] = a_1 p_1 \geq T(1 + p_1) \geq T$.

Assume (3.1) holds for $N = k$.

When $N = K + 1$, we have

$$\begin{aligned} \mathbb{E}[\text{Tr}(S, T)] \geq T &\Leftrightarrow \sum_{i \in S} (a_i - T)p_i \geq T \\ &\Leftrightarrow \sum_{i \in S \setminus \{1\}} (a_i - Q)p_i \geq Q \end{aligned}$$

where $Q = T + \frac{p_1(T - a_1)}{1 + \sum_{i=2}^N p_i} < T$ is a constant number independent on a_2, a_3, \dots, a_N . Hence $a_i \geq Q$ for any i and $\sum_{i \in S \setminus \{1\}} (a_i - Q)p_i = \mathbb{E}[\text{Tr}(S \setminus \{1\}, Q)]$. By the induction hypothesis, $\mathbb{E}[\max_{i \in S \setminus \{1\}} X_i] \geq Q$ and

$$\mathbb{E}[\max_{i \in S} X_i] = a_1 p_1 + (1 - p_1) \mathbb{E}[\max_{i \in S \setminus \{1\}} X_i] \geq a_1 p_1 + (1 - p_1)Q \geq T.$$

Therefore for any integer N , $\mathbb{E}[\text{Tr}(S, T)] \geq T \Rightarrow \mathbb{E}[\max_{i \in S} X_i] \geq T$. ■

With Lemma 3.3.5, now we can prove Theorem 3.3.4.

Say the while loop stopped at T_0 and Algorithm 1 outputs S_0 . According to the description of Algorithm 1,

$$\mathbb{E}[\text{Tr}(S_0, T_0)] > T_0, \tag{3.2}$$

and according to the definition of \mathcal{A} ,

$$\max_{S \in \mathcal{F}} \mathbb{E}[\text{Tr}(S, T_0(1 + \varepsilon))] \leq T_0(1 + \varepsilon)/\alpha. \tag{3.3}$$

(3.2) leads to $\mathbb{E}[\max_{i \in S_0} X_i] \geq T_0$ by Lemma 3.3.5, and (3.3) implies that for any $S \in \mathcal{F}$

$$\mathbb{E}[\max_{i \in S} X_i] \leq T_0(1 + \varepsilon) + \mathbb{E}[\text{Tr}(S, T_0(1 + \varepsilon))] \leq T_0(1 + \varepsilon)(1 + 1/\alpha).$$

Combine them to get

$$\mathbb{E}[\max_{i \in S_0} X_i] \geq \max_{S \in \mathcal{F}} \mathbb{E}[\max_{i \in S} X_i] / ((1 + \varepsilon)(1 + 1/\alpha)) = \frac{\alpha}{(1 + \alpha)(1 + \varepsilon)} \text{OPT}.$$

When the while loop starts, $T = \max a_i$. When $T = \frac{\min a_i}{1 + 1/\min p_i}$, for any $S \neq \emptyset$, $\mathbb{E}[\text{Tr}(S, T)] \geq (\min a_i - T) \min p_i \geq T$, the while loop must have terminated. Hence \mathcal{A} is called by at most $\log_{1+\varepsilon} \frac{(1+1/\min p_i) \max a_i}{\min a_i}$ times, which is polynomial in the input size. As \mathcal{A} itself is a polynomial time algorithm, the total running time of Algorithm 1 is polynomial. ■

3.3.2 General Cases

We provide two methods to handle general cases here. The first one is decomposing a discrete distribution to a set of Bernoulli distributions. It is a little complicated, but this trick will be used again in the design of PTAS. The second one is applying Algorithm 1 immediately.

For the correctness, we will show that Lemma 3.3.5 holds for any distributions.

Method 1: For any discrete distributions, we can rewrite it as the maximum of a set of Bernoulli distributions.

Definition 3.3.6. For a discrete distribution X with supports $0 = a_0 < a_1 \leq a_2 \leq \dots \leq a_l$ and $\Pr[X = a_j] = p_j$, we define a set of independent Bernoulli variables $\{Z_j\}_{j=1}^l$ as

$$Z_j = B(a_j, \frac{p_j}{\sum_{j' \leq j} p_{j'}}).$$

We call $\{Z_j\}$ the Bernoulli decomposition of X .

Lemma 3.3.7. For a discrete distribution X and its Bernoulli decomposition $\{Z_j\}$, $\max_j \{Z_j\}$ has the same distribution with X .

Proof. It follows an easy computing.

$$\begin{aligned} \Pr[\max\{Z_j\} = a_i] &= \Pr[Z_i = a_i] \prod_{h>i} \Pr[Z_h = 0] \\ &= \frac{p_i}{\sum_{i' < i} p_{i'}} \prod_{h>i} \left(1 - \frac{p_h}{\sum_{h' \leq h} p_{h'}}\right) \\ &= \frac{p_i}{\sum_{i' < i} p_{i'}} \prod_{h>i} \frac{\sum_{h' \leq h-1} p_{h'}}{\sum_{h' \leq h} p_{h'}} \\ &= p_i. \end{aligned}$$

Hence $\forall i \leq l$, $\Pr[\max\{Z_j\} = a_i] = \Pr[X = a_i]$. ■

With Lemma 3.3.7, for any set S , we know that $\max_{i \in S} X_i$ and $\max_{i \in S, j} Z_{i,j}$ have the same distribution. As all $Z_{i,j}$ are independent Bernoulli variables, we can apply an algorithm similar to Algorithm 1 on $\{Z_{i,j}\}$ and obtain a constant approximation. The pseudocode can be found in Algorithm 2.

Theorem 3.3.8. For any set of non-negative discrete distributions, if there is an α -approximation for MAX-SUM, there is an $\frac{\alpha}{(1+\alpha)(1+\epsilon)}$ -approximation for MAXIMUM-ELEMENT.

Algorithm 2 Constant Approximation for MAXIMUM-ELEMENT

Require: $(X_1, X_2, \dots, X_n, \mathcal{F})$

- 1: For each X_i compute the Bernoulli decomposition $\{Z_{i,j}\}_j$ of X_i
 - 2: $T \leftarrow \max a_{i,j}$
 - 3: $S \leftarrow \mathcal{A}(\sum_j \mathbb{E}[\text{Tr}(Z_{1,j}, T)], \dots, \sum_j \mathbb{E}[\text{Tr}(Z_{n,j}, T)], \mathcal{F})$
 - 4: **while** $\mathbb{E}[\sum_{i \in S, j} \text{Tr}(Z_{i,j}, T)] \leq T$ **do**
 - 5: $T \leftarrow T/(1 + \varepsilon)$
 - 6: $S \leftarrow \mathcal{A}(\sum_j \mathbb{E}[\text{Tr}(Z_{1,j}, T)], \dots, \sum_j \mathbb{E}[\text{Tr}(Z_{n,j}, T)], \mathcal{F})$
 - 7: **end while**
 - 8: Return S
-

Proof. Take the same notations as in the proof of Theorem 3.3.4. Similar to (3.2) and (3.3), we have

$$\mathbb{E}[\sum_{i \in S_0, j} \text{Tr}(Z_{i,j}, T_0)] \leq T_0, \quad (3.4)$$

and

$$\max_{S \in \mathcal{F}} \sum_{i \in S} \sum_j \mathbb{E}[\text{Tr}(Z_{i,j}, T_0(1 + \varepsilon))] \leq T_0(1 + \varepsilon)/\alpha. \quad (3.5)$$

(3.4) and (3.5) imply that

$$\mathbb{E}[\max_{i \in S_0, j} Z_{i,j}] \geq \frac{\alpha}{(1 + \alpha)(1 + \varepsilon)} \max_{S \in \mathcal{F}} \mathbb{E}[\max_{i \in S, j} Z_{i,j}]. \quad (3.6)$$

By Lemma 3.3.7, we have for any S

$$\max_{i \in S} X_i = \max_{i \in S, j} Z_{i,j}. \quad (3.7)$$

Put (3.7) into (3.6) and we complete the proof. ■

Method 2: As we said, we show a stronger version of Lemma 3.3.5.

Lemma 3.3.9. For any set of discrete distributions $\{X_i\}_{i=1}^n$, if $\sum_i \mathbb{E}[\text{Tr}(X_i, T)] \geq T$,

$$\mathbb{E}[\max_i X_i] \geq T.$$

Proof. Assume $\text{Supp}(X_i) = \{a_{i,j}\}_{j=0}^{s_i}$ with $0 = a_{i,0} < T < a_{i,1} \leq a_{i,2} \leq \dots \leq a_{i,s_i}$. Denote $p_{i,j} = \Pr[X_i = a_{i,j}]$. For each X_i , we create a Bernoulli variable $Y_i = B(u_i, q_i)$ with $u_i = \frac{\sum_{j>0} a_{i,j} p_{i,j}}{\sum_{j>0} p_{i,j}}$ and $q_i = \sum_{j>0} p_{i,j}$. It is easy to check that

$$\sum_i \mathbb{E}[\text{Tr}(Y_i, T)] = \sum_i \mathbb{E}[\text{Tr}(X_i, T)] \geq T.$$

By Lemma 3.3.5, we have $\mathbb{E}[\max_i Y_i] \geq T$. Now we prove that

$$\mathbb{E}[\max_i X_i] \geq \mathbb{E}[\max_i Y_i]. \quad (3.8)$$

If $n = 1$, $\mathbb{E}[X_1] = \mathbb{E}[Y_1]$, (3.8) holds.

Assume (3.8) holds for $n = k$. When $n = k + 1$, assume u_1 is the largest among all u_i s.

On one hand,

$$\begin{aligned} \mathbb{E}[\max_i Y_i] &= u_1 q_1 + (1 - q_1) \mathbb{E}[\max_{i>1} Y_i] \\ &\leq u_1 q_1 + (1 - q_1) \mathbb{E}[\max_{i>1} X_i] \\ &= \mathbb{E}[X_1] + \Pr[X_1 = 0] \mathbb{E}[\max_{i>1} X_i]. \end{aligned}$$

The first inequality follows from the induction hypothesis, and the last equality follows from the definition of u_1, q_1 . On the other hand,

$$\begin{aligned} \mathbb{E}[\max_i X_i] &= \Pr[X_1 = 0] \mathbb{E}[\max_{i>1} X_i] + \Pr[X_1 > 0] \mathbb{E}[\max_i X_i \mid X_1 > 0] \\ &\geq \Pr[X_1 = 0] \mathbb{E}[\max_{i>1} X_i] + \Pr[X_1 > 0] \mathbb{E}[X_1 \mid X_1 > 0] \\ &= \Pr[X_1 = 0] \mathbb{E}[\max_{i>1} X_i] + \mathbb{E}[X_1]. \end{aligned}$$

Hence we have $\mathbb{E}[\max_i X_i] \geq \Pr[X_1 = 0] \mathbb{E}[\max_{i>1} X_i] + \mathbb{E}[X_1] \geq \mathbb{E}[\max_i Y_i]$, which completes the proof. \blacksquare

Using Lemma 3.3.9 instead of Lemma 3.3.5, we know that the approximation guarantee of Algorithm 1 holds for any discrete distributions.

3.4 PTAS

Now we provide a PTAS for MAXIMUM-ELEMENT, taking $\mathcal{F} = \{S \mid |S| = K\}$ as example. In other words, given a constant $\varepsilon \in (0, 0.5)$, we give an algorithm which finds a solution $S \in \mathcal{F}$ and $\mathbb{E}[\max_{i \in S} X_i] \geq (1 - \varepsilon)\text{OPT}$.

We first provide an overview of our approach and explain details later.

- (Discretization) We first run a constant approximation algorithm to obtain an estimation of OPT , and transform each X_i to another distribution \bar{X}_i according to the estimation, such that all \bar{X}_i s are supported on a set of size $O(1/\varepsilon^2)$;
- (Computing signatures) For each X_i , we can compute a signature $\text{Sg}(X_i)$, which is a vector of size $O(1/\varepsilon^2)$, using \bar{X}_i . For any set S , we also define a signature $\text{Sg}(S) = \sum_{i \in S} \text{Sg}(X_i)$. We will show that two sets with the same signature must have close objective values.
- (Enumeration) We enumerate all possible signatures which is polynomial and enumerable. For each signature, try to find one set with this signature and its cardinality being K . Finally output the found set with the maximum objective value.

3.4.1 Discretization

We first explain details in discretization. The pseudocode for discretization can be found in Algorithm 3.

We run Algorithm 1 to obtain a solution S_G and denote $W = \mathbb{E}[\max_{i \in S_G} X_i]$. When $\mathcal{F} = \{S \mid |S| = K\}$, MAX-SUM can be solved with $\alpha = 1$ and by Theorem 3.3.8 $W \geq c\text{OPT}$

Algorithm 3 Discretization

```
1: We first run Algorithm 1 to obtain a solution  $S_G$  and let  $W = \mathbb{E}[\max_{i \in S_G} X_i]$ .
2: for  $i = 1$  to  $n$  do
3:   Compute the Bernoulli decomposition  $\{Z_{i,j}\}_j$  of  $X_i$ .
4:   for all  $Z_{i,j}$  do
5:     Create another Bernoulli variable  $\bar{Z}_{i,j}$  as follows:
6:     if  $a_{i,j} > W/\varepsilon$  then
7:       Let  $\bar{Z}_{i,j} \sim B\left(\frac{W}{\varepsilon}, \mathbb{E}[Z_{i,j}]\frac{\varepsilon}{W}\right)$ ; (Case 1)
8:     else
9:       Let  $\bar{Z}_{i,j} = \lfloor \frac{Z_{i,j}}{\varepsilon W} \rfloor \varepsilon W$ ; (Case 2)
10:    end if
11:  end for
12:  Let  $\bar{X}_i = \max_j \{\bar{Z}_{i,j}\}$ ;
13: end for
```

with $c = \frac{1}{2(1+\varepsilon)}$. We first compute the Bernoulli decomposition $\{Z_{i,j}\}$ of X_i . For each $Z_{i,j}$ we create $\bar{Z}_{i,j}$ for it as follows: We distinguish two cases according to $a_{i,j}$, the nonzero possible value of $Z_{i,j}$:

- Case 1: $a_{i,j} > W/\varepsilon$, we let $\bar{Z}_{i,j} = B(W/\varepsilon, \mathbb{E}[Z_{i,j}]\varepsilon/W)$;
- Case 2: $a_{i,j} \leq W/\varepsilon$, we let $\bar{Z}_{i,j} = \lfloor \frac{Z_{i,j}}{\varepsilon W} \rfloor \varepsilon W$.

Finally, we let $\bar{X}_i = \max_j \bar{Z}_{i,j}$, which is the discretization of X_i . Note that all $\bar{Z}_{i,j}$ s are supported on $Supp = \{0, \varepsilon W, 2\varepsilon W, \dots, W/\varepsilon\}$, and so \bar{X}_i is also a discrete distribution supported on $Supp$. We can easily compute $\Pr[\bar{X}_i = s]$ for any $s \in Supp$.

We show that our discretization only include a small loss in the objective value. The key is to show that we don't lose much in the transform from Z to \bar{Z} , which is the only one step producing difference.

Lemma 3.4.1. *Consider any set of Bernoulli variables $\{Z_i = B(a_i, p_i)\}_{i=1}^m$. Assume that $\mathbb{E}[\max_{i \in [m]} Z_i] < cW$, where c is a constant such that $c\varepsilon < 1/2$. For each Z_i we create \bar{Z}_i in the same way as Algorithm 3. The following holds:*

$$\mathbb{E}[\max Z_i] \geq \mathbb{E}[\max \bar{Z}_i] \geq \mathbb{E}[\max Z_i] - (2c^2 + 1)\varepsilon W. \quad (3.9)$$

Proof. Assume a_1 is the largest among a_i s.

If $a_1 \leq W/\varepsilon$, all transformation is in Case 2. In this case, it is obvious to have

$$\mathbb{E}[\max Z_i] \geq \mathbb{E}[\max \bar{Z}_i] \geq \mathbb{E}[\max Z_i] - \varepsilon W \geq \mathbb{E}[\max Z_i] - (2c + 1)\varepsilon W.$$

If $a_1 > W/\varepsilon$, we do induction on m . Let $L = \{i | a_i > W/\varepsilon\}$. We prove the following claim:

$$\mathbb{E}[\max Z_i] \geq \mathbb{E}[\max \bar{Z}_i] \geq \mathbb{E}[\max Z_i] - \varepsilon W - c\varepsilon \sum_{i \in L} a_i p_i. \quad (3.10)$$

When $m = 1$, (3.10) holds immediately as $\mathbb{E}[Z_1] = \mathbb{E}[\bar{Z}_1]$ in Case 1.

Assume (3.10) holds for $m = k$, we show it holds for $m = k+1$. As $\bar{Z}_1 = B(W/\varepsilon, \mathbb{E}[Z_1]\varepsilon/W)$, we have:

$$\begin{aligned} \mathbb{E}[\max_{i \geq 1} Z_i] - \mathbb{E}[\max_{i \geq 1} \bar{Z}_i] &= a_1 p_1 + (1 - p_1) \mathbb{E}[\max_{i \geq 2} Z_i] - a_1 p_1 - (1 - \mathbb{E}[Z_1]\varepsilon/W) \mathbb{E}[\max_{i \geq 2} \bar{Z}_i] \\ &\geq (1 - p_1) \mathbb{E}[\max_{i \geq 2} \bar{Z}_i] - (1 - \mathbb{E}[Z_1]\varepsilon/W) \mathbb{E}[\max_{i \geq 2} \bar{Z}_i] \\ &= (\varepsilon a_1/W - 1) p_1 \mathbb{E}[\max_{i \geq 2} \bar{Z}_i] \geq 0 \end{aligned}$$

where the first inequality follows from induction hypothesis and the last comes from $a_1 > W/\varepsilon$. The second inequality in 3.10 can be seen as follows:

$$\begin{aligned} \mathbb{E}[\max_{i \geq 1} \bar{Z}_i] - \mathbb{E}[\max_{i \geq 1} Z_i] &= a_1 p_1 + (1 - \mathbb{E}[Z_1]\varepsilon/W) \mathbb{E}[\max_{i \geq 2} \bar{Z}_i] - a_1 p_1 - (1 - p_1) \mathbb{E}[\max_{i \geq 2} Z_i] \\ &\geq (p_1 - \mathbb{E}[Z_1]\varepsilon/W) \mathbb{E}[\max_{i \geq 2} Z_i] - \varepsilon W - c\varepsilon \sum_{i \in L \setminus \{1\}} a_i p_i \\ &\geq -\varepsilon W - c\varepsilon \sum_{i \in L} a_i p_i, \end{aligned}$$

where the first inequality follows from induction hypothesis and the last comes from $p_1 \geq 0$ and $\mathbb{E}[\max_{i \geq 2} Z_i] \leq cW$. Proof of 3.10 finishes.

Finally, we show $\sum_{i \in L} a_i p_i \leq 2cW$. According to Markov inequality, we have

$$\Pr[\max Z_i \geq W/\varepsilon] \leq c\varepsilon,$$

which is equivalent to

$$\prod_{i \in L} (1 - p_i) \geq 1 - \varepsilon.$$

Then we can see that

$$cW \geq \mathbb{E}[\max_{i \in L} Z_i] = \sum_{i \in L} a_i p_i \prod_{j < i} (1 - p_j) \geq (1 - c\varepsilon) \sum_{i \in L} a_i p_i \geq \sum_{i \in L} a_i p_i / 2.$$

Plugging this into (3.10), we prove the lemma. ■

Corollary 3.4.2. *For any set S , suppose that $\mathbb{E}[\max_{i \in S} X_i] < cW$ where c is a constant such that $c\varepsilon < 1/2$, we create \bar{X}_i as in Algorithm 3, and the following holds:*

$$\mathbb{E}[\max_{i \in S} X_i] \geq \mathbb{E}[\max_{i \in S} \bar{X}_i] \geq \mathbb{E}[\max_{i \in S} X_i] - (2c^2 + 1)\varepsilon W.$$

3.4.2 Signatures

For each X_i , we have created its discretization \bar{X}_i . Since \bar{X}_i is a discrete distribution, we compute its Bernoulli decomposition $\{Y_{i,j}\}_{j=1}^h$ where $h = |\text{Supp}|$. Note that since $\{X_i\}$ and $\{\bar{X}_i\}$ have different supports, $\{Y_{i,j}\}$ is different from $\{\bar{Z}_{i,j}\}$ which is defined in the discretization step. Suppose $Y_{i,j} = B(jW/\varepsilon, q_{i,j})$. We define the signature of X_i to be

$$\text{Sg}(X_i) = (\lfloor \frac{-\log(1 - q_{i,1})}{\varepsilon^4/n} \rfloor \varepsilon^4/n, \dots, \lfloor \frac{-\log(1 - q_{i,h})}{\varepsilon^4/n} \rfloor \varepsilon^4/n). \quad (3.11)$$

And for any set S we define its signature to be

$$\text{Sg}(S) = \sum_{i \in S} \text{Sg}(X_i).$$

Define the set of signature vectors \mathbf{SG} to be all h -dimensional vectors with each coordinate being an integer multiples of ε^4/n and at most $\log(h/\varepsilon^2)$.

For any signature vector \mathbf{sg} , we associate a set of random variables $\{B_k = B(k\varepsilon\mathbf{W}, 1 - e^{-\mathbf{sg}_k})\}_{k=1}^h$ to it. It is easy to see that the signature of $\max B_k$ is exactly \mathbf{sg} . And we define the value of \mathbf{sg} to be $\text{Val}(\mathbf{sg}) = \mathbb{E}[\max B_k]$.

Lemma 3.4.3. *Consider two sets S_1 and S_2 . If $\mathbf{Sg}(S_1) = \mathbf{Sg}(S_2)$, the following holds:*

$$|\mathbb{E}[\max_{i \in S_1} \bar{X}_i] - \mathbb{E}[\max_{i \in S_2} \bar{X}_i]| \leq O(\varepsilon)\mathbf{W}.$$

Proof. Suppose $\{Y_{i,j}\}_{j=1}^h$ is the Bernoulli decomposition of \bar{X}_i . For any set S , define $Y_k(S) = \max_{i \in S} Y_{i,k}$. It is easy to see that $Y_k(S)$ is a Bernoulli variable $B(k\varepsilon\mathbf{W}, p_k(S))$ with $p_k(S) = 1 - \prod_{i \in S} (1 - q_{i,k})$. As $\mathbf{Sg}(S_1) = \mathbf{Sg}(S_2)$, we have

$$|p_k(S_1) - p_k(S_2)| \leq 2\varepsilon^4 \forall k \in [h].$$

Note that $\max_{i \in S} \bar{X}_i = \max_k Y_k(S)$, we have

$$\begin{aligned} |\mathbb{E}[\max_{i \in S_1} \bar{X}_i] - \mathbb{E}[\max_{i \in S_2} \bar{X}_i]| &= |\mathbb{E}[\max_k Y_k(S_1)] - \mathbb{E}[\max_k Y_k(S_2)]| \\ &\leq \mathbf{W}/\varepsilon \left(\sum_k |p_k(S_1) - p_k(S_2)| \right) \\ &\leq 4h\varepsilon^3\mathbf{W} = O(\varepsilon)\mathbf{W} \end{aligned}$$

where the first inequality follows from the following lemma.

Lemma 3.4.4. *[6] Let $P = P_1 \times P_2 \times \cdots \times P_n$ and $Q = Q_1 \times \cdots \times Q_n$ be two disctributions, then we have*

$$L_1(P, Q) \leq \sum L_1(P_i, Q_i).$$

■

Corollary 3.4.5. *For any feasible set S with $\mathbf{Sg}(S) = \mathbf{sg}$, $|\mathbb{E}[\max_{i \in S} \bar{X}_i] - \text{Val}(\mathbf{sg})| \leq O(\varepsilon)\mathbf{W}$.*

Combined with Corollary 3.4.2, we have that $|\mathbb{E}[\max_{i \in S} X_i] - \text{Val}(\mathbf{sg})| \leq O(\varepsilon)W$.

As we only consider signatures whose coordinates are all at most $\log h/\varepsilon^2$, the number of different signatures is $O((n \log \frac{h}{\varepsilon^2}/\varepsilon^4)^{h-1})$, which is polynomial. And for any signature \mathbf{sg} with coordinates larger than $\log(h/\varepsilon^2)$, consider the signature \mathbf{sg}' with those large coordinates decreased to exact $\log(h/\varepsilon^2)$. It is easy to show that $|\text{Val}(\mathbf{sg}) - \text{Val}(\mathbf{sg}')| \leq \varepsilon W$ and by Corollary 3.4.5, we lose at most $O(\varepsilon)W$ by ignoring those signatures.

3.4.3 Enumerating Signatures

Our algorithm enumerates all signature vectors \mathbf{sg} in SG . For each \mathbf{sg} , we check if we can find a set S of size K such that $\text{Sg}(S) = \mathbf{sg}$.

Take the cardinality constraint as example, this can be done by a standard dynamic program in $\tilde{O}(m^{O(1/\varepsilon^2)})$ time as follows: We use Boolean variable $R[i][j][\mathbf{sg}']$ to represent whether signature vector $\mathbf{sg}' \in \text{SG}$ can be dominated by i variables in set $\{X_1, \dots, X_j\}$. The dynamic programming recursion is

$$R[i][j][\mathbf{sg}'] = R[i][j-1][\mathbf{sg}'] \wedge R[i-1][j-1][\mathbf{sg}' - \text{Sg}(X_j)].$$

If the answer is yes (i.e., we can find such S), we say \mathbf{sg} is a feasible signature vector and S is a candidate set.

For a general constraint, as each coordinate of one signature is an integer multiple of ε^4/n with upper-bound $\log(h/\varepsilon^2)$, we can encode each signature to an integer with upper-bound $O((n \log \frac{h}{\varepsilon^2}/\varepsilon^4)^{h-1})$. Thus finding a corresponding feasible set reduced to a EXACT-SUM problem, and if there is one pseudo-polynomial algorithm for EXACT-SUM, we can find a feasible set in polynomial time.

Finally, we pick the candidate set with maximum $\mathbb{E}[\max_{i \in S} X_i]$ and output the set. The pseudocode can be found in Algorithm 4.

Now we are ready to prove Theorem 3.4.6 by showing that Algorithm 4 is a PTAS for MAXIMUM-ELEMENT.

Algorithm 4 PTAS for MAXIMUM-ELEMENT

- 1: Run Algorithm 3 to obtain random variables $\{\bar{X}_i\}$;
- 2: Compute Bernoulli decomposition $\{Y_{i,j}\}_1^h$ for each \bar{X}_i ;
- 3: Compute signatures of $\{X_i\}$ according to (3.11);
- 4: $U \leftarrow \emptyset$
- 5: **for all** signature vector $\mathbf{sg} \in \mathbf{SG}$ **do**
- 6: Find a set S such that $|S| = K$ and $\mathbf{Sg}(S) = \mathbf{sg}$;
- 7: **if** $\mathbb{E}[\max_{i \in S} X_i] > \mathbb{E}[\max_{i \in U} X_i]$ **then**
- 8: $U \leftarrow S$.
- 9: **end if**
- 10: **end for**
- 11: Return U

Theorem 3.4.6. *Algorithm 4 is a PTAS for MAXIMUM-ELEMENT, if there is a pseudopolynomial time algorithm for EXACT-SUM.*

Proof. Suppose S^* is the optimal solution and \mathbf{sg}^* is the signature of S^* . By Corollary 3.4.5, we have that $|\text{OPT} - \text{Val}(\mathbf{sg}^*)| \leq O(\varepsilon)W$.

When Algorithm 4 is enumerating \mathbf{sg}^* , it can find a set S such that $\mathbf{Sg}(S) = \mathbf{sg}^*$ (there exists at least one such set since S^* is one). Therefore, we can see that

$$|\mathbb{E}[\max_{i \in S} X_i] - \mathbb{E}[\max_{i \in S^*} X_i]| \leq |\text{Val}(\mathbf{sg}^*) - \mathbb{E}[\max_{i \in S} X_i]| + |\text{Val}(\mathbf{sg}^*) - \mathbb{E}[\max_{i \in S^*} X_i]| \leq O(\varepsilon)W.$$

Let U be the output of Algorithm 4. Since $W \geq c\text{OPT}$, we have $\mathbb{E}[\max_{i \in U} X_i] \geq \mathbb{E}[\max_{i \in S} X_i] \geq (1 - O(\varepsilon))\text{OPT}$.

If there is a pseudopolynomial time algorithm for EXACT-SUM, the running time of the algorithm is polynomial for a fixed constant $\varepsilon > 0$. Hence, we have a PTAS problem. ■

3.5 Minimization version of maximum-element

In this part we consider the minimization version of MAXIMUM-ELEMENT. For the constant approximation, we propose a similar algorithm with Algorithm 1, whose pseudocode can be found in Algorithm 5, where \mathcal{A} is an α -approximation for MIN-SUM problem.

Algorithm 5 Constant Approximation for Minimization

Require: $(a_1, a_2, \dots, a_n, p_1, p_2, \dots, p_n, \mathcal{F})$

- 1: $T \leftarrow \min a_i \min p_i / (\alpha + \min p_i)$
 - 2: $S \leftarrow \mathcal{A}(\mathbb{E}[\text{Tr}(X_1, T)], \dots, \mathbb{E}[\text{Tr}(X_n, T)], \mathcal{F})$
 - 3: **while** $\mathbb{E}[\text{Tr}(S, T)] \geq \alpha T$ **do**
 - 4: $T \leftarrow T(1 + \varepsilon)$
 - 5: $S \leftarrow \mathcal{A}(\mathbb{E}[\text{Tr}(X_1, T)], \dots, \mathbb{E}[\text{Tr}(X_n, T)], \mathcal{F})$
 - 6: **end while**
 - 7: Return S
-

Theorem 3.5.1. *Algorithm 5 returns an output with approximation ratio $\frac{1+\alpha}{1+\varepsilon}$ in polynomial time.*

The proof of Theorem 3.5.1 is almost the same as the proof of Theorem 3.3.4, and we skip it here.

For the PTAS approach, we still use the same discretization and signature as in solving MAXIMUM-ELEMENT. Obviously, the constant approximation can be obtained by Algorithm 5 instead of Algorithm 1. Though the second inequality of (3.9) no longer holds as condition $\mathbb{E}[\max_{i \in S} Z_i] \leq cW$ no longer holds for any constant c , we show the following lemma which is enough for the minimization version.

Lemma 3.5.2. *For any set of Bernoulli variables $\{Z_i\}$ with $\mathbb{E}[Z_i] \leq W$ and $\mathbb{E}[\max Z_i] > W/3\varepsilon$, the following holds:*

$$\mathbb{E}[\max \bar{Z}_i] \geq CW/3\varepsilon - \varepsilon W,$$

where $(1 - C/3)(1 - C) = C$.

Proof. Define $S_1 = \{i | a_i > W/\varepsilon\}$ and $S_2 = \{i | a_i \leq W/\varepsilon\}$.

If $\mathbb{E}[\max_{i \in S_2} Z_i] \geq CW/3\varepsilon$, we have

$$\mathbb{E}[\max_{i \in S} \bar{Z}_i] \geq \mathbb{E}[\max_{i \in S_2} \bar{Z}_i] \geq CW/3\varepsilon - \varepsilon W.$$

If $\mathbb{E}[\max_{i \in S_2} Z_i] < CW/3\varepsilon$, we have

$$\mathbb{E}[\max_{i \in S_1} Z_i] \geq \mathbb{E}[\max_{i \in S} Z_i] - \mathbb{E}[\max_{i \in S_2} Z_i] \geq (1 - C)W/3\varepsilon.$$

Therefore we take $T = CW/3\varepsilon$ and have

$$\begin{aligned}
\sum_{i \in S_1} \mathbb{E}[\text{Tr}(\bar{Z}_i, T)] &= \sum_{i \in S_1} (W/\varepsilon - T) a_i p_i \varepsilon / W \\
&= \sum_{i \in S_1} (1 - C/3) a_i p_i \\
&\geq (1 - C/3) \mathbb{E}[\max_{i \in S_1} Z_i] \\
&\geq (1 - C/3)(1 - C)W/3\varepsilon \\
&= CW/3\varepsilon = T.
\end{aligned}$$

By Lemma 3.3.5, we have $\mathbb{E}[\max_{i \in S_1} \bar{Z}_i] \geq T$ and so

$$\mathbb{E}[\max_{i \in S} \bar{Z}_i] \geq \mathbb{E}[\max_{i \in S_1} \bar{Z}_i] \geq T \geq CW/3\varepsilon - \varepsilon W.$$

■

Corollary 3.5.3. *For any set S , suppose that $\mathbb{E}[\max_{i \in S} X_i] \geq W/3\varepsilon$, the following holds:*

$$\mathbb{E}[\max_{i \in S} \bar{Z}_i] \geq CW/3\varepsilon - \varepsilon W,$$

where $(1 - C/3)(1 - C) = C$.

With Corollary 3.4.2 and Corollary 3.5.3, we are now ready to prove our PTAS result.

Theorem 3.5.4. *The optimal solution with set $\{\bar{X}_i\}$ is a $(1 + O(\varepsilon))$ -approximation to the optimal solution with set $\{X_i\}$. So Algorithm 6 is a PTAS for minimization version of MAXIMUM-ELEMENT, if there is a pseudopolynomial time algorithm for EXACT-SUM.*

Proof. Assume the optimal solution for $\{X_i\}$ is S and the optimal solution for $\{\bar{X}_i\}$ is \bar{S} .

As \bar{S} is optimal for $\{\bar{X}_i\}$, we have

$$\mathbb{E}[\max_{i \in \bar{S}} \bar{X}_i] \leq \mathbb{E}[\max_{i \in S} \bar{X}_i] \leq \mathbb{E}[\max_{i \in S} X_i] = \text{OPT}, \quad (3.12)$$

Algorithm 6 PTAS for Minimization

- 1: Run Algorithm 3 with Algorithm 5 instead of Algorithm 1 to obtain $\{\bar{X}_i\}$;
 - 2: Compute signatures according to (3.11);
 - 3: $U \leftarrow \emptyset$;
 - 4: **for all** signature vector $\mathbf{sg} \in \mathbf{SG}$ **do**
 - 5: Find a set S such that $|S| = K$ and $\mathbf{Sg}(S) = \mathbf{sg}$;
 - 6: **if** $\mathbb{E}[\max_{i \in S} X_i] < \mathbb{E}[\max_{i \in U} X_i]$ **then**
 - 7: $U \leftarrow S$.
 - 8: **end if**
 - 9: **end for**
 - 10: Return U
-

the last inequality follows from Corollary 3.4.2. On the other hand, say $\mathbb{E}[\max_{i \in \bar{S}} X_i] = kW$.

If $k > 1/3\varepsilon$, by Corollary 3.5.3, $\mathbb{E}[\max_{i \in \bar{S}} X_i] \geq CW/3\varepsilon - \varepsilon W$. Recall that W is an constant approximation to the optimal solution, and this contradicts to 3.12.

If $k \leq 1/3\varepsilon$, as $k\varepsilon < 1/2$, we apply Corollary 3 and obtain

$$\mathbb{E}[\max_{i \in \bar{S}} \bar{X}_i] \geq kW - (2k^2 + 1)\varepsilon W. \quad (3.13)$$

Combine 3.13 and 3.12 we have $[k - (2k^2 + 1)\varepsilon]W \leq \text{OPT}$. Solve it and get the solution $kW \leq (1 + O(\varepsilon))\text{OPT}$ (ignore the other solution of this inequality which implies $k > 1/3\varepsilon$). This finishes the proof of the first part of this theorem.

Almost same as the proof for Theorem 3.4.6, it is obvious that Algorithm 6 finds a solution \tilde{S} such that $\mathbb{E}[\max_{i \in \tilde{S}} X_i] \leq (1 + O(\varepsilon))\mathbb{E}[\max_{i \in \bar{S}} X_i]$. And by the first part of this theorem, we have that $\mathbb{E}[\max_{i \in \tilde{S}} X_i] \leq (1 + O(\varepsilon))\mathbb{E}[\max_{i \in S} X_i]$. ■

3.6 MAX_K

In many applications, people consider the K -th largest value instead of the largest. For example, in the second price auction, the final price of an item is the second largest bid. For this purpose, we define the MAX_K function and the MAX_K problem.

Definition 3.6.1. *Given a set of real numbers $\{v_i\}_{i=1}^n$, an integer $K \geq 1$, and set $S \subset [n]$,*

define

$$\text{MAX}_K(S) = \text{the } K\text{-th largest value in set } \{v_i \mid i \in S\}.$$

Problem 3.6.2 (MAX_K). *Inputs: a set of non-negative distributions $\{X_i\}_{i=1}^n$, an integer $K \geq 1$, and a constraint $\mathcal{F} \subset 2^{[n]}$;*

Outputs: a set $S \in \mathcal{F}$, which maximizes the value $\mathbb{E}[\text{MAX}_K(S)]$.

If $|S| = K$, it is obvious that $\text{MAX}_K(S) = \min_{i \in S} v_i$. So for a general K and constraint $\mathcal{F} = \{S \mid |S| = K\}$, the MAX_K problem can be as difficult as the MINIMUM-ELEMENT problem. As we will show in Chapter 4, it is impossible to approximate to any polynomial approximation ratio. To extend our result to MAX_K problem, we try to bound the top function by the maximum of a set of independent random variables. We prove the following upper-bound lemma and leave the other side as a conjecture.

Lemma 3.6.3. *Consider a set of Bernoulli variables $\{X_{i,j}\}_{i,j=1}^n$. For any i , $X_{i,1}, X_{i,2}, \dots, X_{i,n}$ are identically independent distributions. The following holds:*

$$\Pr[\max_{i < j} \min(X_{i,j}, X_{j,i}) = 1] \geq \Pr[\max_{i < j} \min(X_{i,i}, X_{j,j}) = 1]. \quad (3.14)$$

Conjecture: The following holds for some constant C :

$$\Pr[\max_{i < j} \min(X_{i,i}, X_{j,j}) = 1] \leq C \Pr[\max_{i < j} \min(X_{i,j}, X_{j,i}) = 1].$$

proof sketch. To prove inequality 3.14, we can prove a more general inequality:

$$\Pr[\max_{(i,j) \in S} \min(X_{i,j}, X_{j,i}) = 1] \geq \Pr[\max_{(i,j) \in S} \min(X_{i,i}, X_{j,j}) = 1], \quad (3.15)$$

where S is any set of pairs.

To prove this, do inductions on the size of S . When adding a pair (s, t) to S , if there is

no s or t appearing in S , the induction is obviously easy. If s or t appears in S , we have

$$\begin{aligned}
& \Pr\left[\max_{(i,j) \in S \cup \{(s,t)\}} \min(X_{i,j}, X_{j,i}) = 1\right] \\
&= \Pr[\min(X_{s,t}, X_{t,s}) = 1] + \Pr[\min(X_{s,t}, X_{t,s}) = 0] \Pr\left[\max_{(i,j) \in S} \min(X_{i,j}, X_{j,i}) = 1 \mid \min(X_{s,t}, X_{t,s}) = 0\right] \\
&= \Pr[\min(X_{s,s}, X_{t,t}) = 1] + \Pr[\min(X_{s,s}, X_{t,t}) = 0] \Pr\left[\max_{(i,j) \in S} \min(X_{i,j}, X_{j,i}) = 1 \mid \min(X_{s,t}, X_{t,s}) = 0\right]
\end{aligned}$$

By carefully analyzing the conditional probability, we can have

$$\begin{aligned}
& \left[\max_{(i,j) \in S} \min(X_{i,j}, X_{j,i}) = 1 \mid \min(X_{s,t}, X_{t,s}) = 0\right] \\
& \geq \Pr\left[\max_{(i,j) \in S} \min(X_{i,i}, X_{j,j}) = 1 \mid \min(X_{s,s}, X_{t,t}) = 0\right].
\end{aligned}$$

Combine them and we have proved 3.15. Lemma 3.6.3 is a special case for $S = \{(i, j) \mid i < j\}$. ■

Theorem 3.6.4. *Based on Conjecture 3.6.3, there is a constant approximation for MAX_2 problem.*

Proof. W.l.o.g., assume all input distributions are supported on integers. Hence we have

$$\mathbb{E}[MAX_2(S)] = \mathbb{E}\left[\max_{i < j} \min(X_i, X_j)\right] = \sum_{t \geq 2} \Pr\left[\max_{i < j} \min(X_i, X_j) \geq t\right].$$

For X_i , we create a set of random variables $\{X_{i,j}\}_{j=1}^n$, identically independent and following the same distribution of X_i . And for X_i and an integer t , we create a set of random variables $\{Y_{i,j}^t\}_{j=1}^n$. They are identically independent and follow the Bernoulli distribution $B(1, \Pr[X_i \geq t])$. It is easy to see that $\Pr[\max_{i < j} \min(X_i, X_j) \geq t] = \Pr[\max_{i < j} \min(Y_{i,i}^t, Y_{j,j}^t) = 1]$. Apply Conjecture 3.6.3 to $\{Y_{i,j}^t\}$ we have

$$\Pr\left[\max_{i < j} \min(Y_{i,j}^t, Y_{j,i}^t) = 1\right] \geq \Pr\left[\max_{i < j} \min(Y_{i,i}^t, Y_{j,j}^t) = 1\right] \leq C \Pr\left[\max_{i < j} \min(Y_{i,j}^t, Y_{j,i}^t) = 1\right].$$

Take summation over all ts we have

$$\mathbb{E}[\max_{i < j} \min(X_{i,j}, X_{j,i})] \geq \mathbb{E}[\text{MAX}_2(S)] \geq C\mathbb{E}[\max_{i < j} \min(X_{i,j}, X_{j,i})],$$

which means that $\mathbb{E}[\text{MAX}_2(S)]$ is constant bounded by $\mathbb{E}[\max_{i < j} \min(X_{i,j}, X_{j,i})]$. Note that $\{\min(X_{i,j}, X_{j,i})\}_{i < j}$ is a set of independent random variables, and by Theorem 3.3.4, Algorithm 1 provides a constant approximation, which is also a constant approximation to MAX_2 problem. ■

The running time is a big problem of Theorem 3.6.4. Applying Algorithm 1 to $\{\min(X_{i,j}, X_{j,i})\}_{i < j}$, the constraint is ‘squared’. For example, the simplest cardinality constraint, finding a maximum set with size k becomes finding a maximum clique with size k in a weighted complete graph. This ‘square’ fact makes us hard to find a natural constraint under which Theorem 3.6.4 provides a polynomial time result.

Chapter 4

MINIMUM-ELEMENT

We formulate MINIMUM-ELEMENT in Chapter 4. We first review the previous technique and present how we combine it and the discretization technique to obtain a better approximation ratio.

4.1 Problem Formulation

We give the formulation of MINIMUM-ELEMENT here.

Problem 4.1.1 (MINIMUM-ELEMENT). *Inputs: a set of independent non-negative random variables $\{X_i\}_{i=1}^n$, costs $\{c_i\}$, and a budget C ;*

Outputs: a set S with total cost at most C , which minimizes the value $\mathbb{E}[\min_{i \in S} X_i]$.

Here X_i follows the discrete distribution D_i with support $\{v_{i,j}\}_{j=1}^{s_i}$ ($s_i = \text{poly}(n)$). Let $p_{i,j} = \Pr[X_i = v_{i,j}]$. The inputs of MINIMUM-ELEMENT are $\{v_{i,j}\}$, $\{p_{i,j}\}$, $\{c_i\}$ and C .

4.2 Hardness

Goel et al. [10] first considered the MINIMUM-ELEMENT problem and proved its hardness.

Theorem 4.2.1. [10] *It is NP-hard to obtain any polynomial approximation for MINIMUM-ELEMENT while preserve the budget, even when all variables share the same cost.*

As it is hard to approximate while preserving the budget, Goel et al. changed the target to minimizing the cost while the objective value is at most $(1 + \varepsilon)\text{OPT}$. The following Theorem 4.2.2 and Theorem 4.2.3 are the foundations of their approach. All we need to do is to find a good initial solution and apply Theorem 4.2.3.

Theorem 4.2.2. [10] Denote $f(S) = \mathbb{E}[\min_{i \in S} X_i]$, then $-\log f$ is a sub-modular function.

Theorem 4.2.3. [20] Given a non-decreasing submodular function f on set U , where each element has a cost, and a budget C . Let $S^* = \text{argmax}\{f(S) \mid \sum_{i \in S} c_i \leq C\}$. With an initial set S , the greedy algorithm using extra cost at most $C \log \frac{f(S^*) - f(S)}{\varepsilon}$ finds a set T such that $f(S^*) - f(T) \leq \varepsilon$.

4.3 CIP Approach

Here we first review the technique used in [10]. Based on their approach we improve their result in the case where all distributions are supported on $\{0, 1, \dots, m-1\}$. The key idea of this approach is to design proper Covering Integer Programs and solve them approximately.

Consider the supports of $\{X_i\}_{i=1}^n$ are $0 = a_0 \leq a_1 \leq a_2 \leq \dots \leq a_m$. Let $l_j = a_{j+1} - a_j$. Recall that $p_{ij} = \Pr[X_i = a_j]$ and let $q_{i,j} = -\log p_{i,j}$. There are two CIPs in their approach: The first one is

$$\begin{aligned}
 & \min . z \\
 & \text{s.t. } \sum_i a_{i,j} y_i \geq \log l_j - z && j = 1, 2, \dots, m \\
 & \sum_i c_i y_i \leq C && \\
 & y_i \in \{0, 1\} && i = 1, 2, \dots, n.
 \end{aligned} \tag{4.1}$$

CIP(4.1) provides a bound for the optimal value when preserve the cost. Assume z^* is the

optimal solution of the first CIP, and the second one is

$$\begin{aligned}
& \min \sum_i c_i y_i \\
& \text{s.t. } \sum a_{i,j} y_i \geq \log l_j - z^* & j = 1, 2, \dots, m \\
& y_i \in \{0, 1\} & i = 1, 2, \dots, n.
\end{aligned} \tag{4.2}$$

CIP(4.2) provides a solution with least cost whose objective value is polynomial bounded by z^* .

In [10] Goel et al. proved the following result by solving these two CIP approximately.

Theorem 4.3.1. *[10] If all X_i s are supported on m values, there is an algorithm providing a solution for MINIMUM-ELEMENT with cost at most $O(\log m)C$ and the objective value at most $m\text{OPT}$. Applying Theorem 4.2.3, there is an algorithm achieves a $O(\log m)$ approximation on cost with objective value at most $(1 + \varepsilon)\text{OPT}$.*

When $\{X_i\}$ are supported on $\{0, 1, \dots, m - 1\}$, we can simply contract intervals and create a set of random variables supported on $\{0, 1, 2, 4, \dots, 2^{\log m}\}$ to approximate and the loss ratio to objective value is at most 2. Hence applying Theorem 4.3.1 in this set leads to Corollary 4.3.2.

Corollary 4.3.2. *[10] When X_i s are supported on $\{0, 1, \dots, m - 1\}$, there is an algorithm achieves an $O(\log \log m)$ approximation on the cost with objective value being at most $(1 + \varepsilon)\text{OPT}$.*

4.4 Our CIP Approach

We consider the special case where $\{X_i\}$ are supported on $[0, m - 1]$. We propose an algorithm to find a $(c, O(\log \log \log m))$ -approximation with some constant c and apply Theorem 4.2.3 to obtain the final solution. Here is our main theorem for MINIMUM-ELEMENT.

Theorem 4.4.1. *When all X_i s are supported on $\{0, 1, \dots, m - 1\}$, there is a polynomial time algorithm providing a $(1 + \varepsilon)$ -approximation with cost at most $O(\log \log \log m)C$.*

We provide an overview of our algorithm here and explain details later.

- Enumerate all possible objective value W ;
- For each W , define a set of possible signatures and enumerate all possible signatures;
- For each signature, solve a corresponding integer programming and find a candidate set;
- Choose a good set among all candidate sets.

As $\mathbb{E}[\min_{i \in S} X_i]$ must fall in the interval $[\min \Pr[X_i \neq 0], m - 1]$. We assume the optimal value being $\{b, 2b, 4b, 8b, \dots, 2^N b\}$ with $lb = \min \Pr[X_i \neq 0]$ and $N = \log_2 \frac{m-1}{b}$. Hence we can enumerate it.

For a guessing W , we need to enumerate signatures. To define signatures, we first introduce a lemma.

Lemma 4.4.2. *Denote $p_{ij} = \Pr[X_i \geq 2^j]$, $j = 0, 1, \dots, \log m$. Then the following holds:*

$$\mathbb{E}[\min_{i \in S} X_i] \leq \sum_{j=0}^{\log m} 2^j \prod_{i \in S} p_{i,j} \leq 2\mathbb{E}[\min_{i \in S} X_i].$$

Proof. Since $\{X_i\}$ are supported on $[0, m - 1]$, we have

$$\begin{aligned} \mathbb{E}[\min_{i \in S} X_i] &= \sum_{k=1}^{m-1} \Pr[\min_{i \in S} X_i \geq k] \\ &= \sum_{j=0}^{\log m} \left(\sum_{k=2^j}^{2^{j+1}-1} \Pr[\min_{i \in S} X_i \geq k] \right) \\ &\leq \sum_{j=0}^{\log m} 2^j \Pr[\min_{i \in S} X_i \geq 2^j] \\ &= \sum_{j=0}^{\log m} 2^j \prod_{i \in S} p_{i,j} \end{aligned}$$

Similarly, we have

$$\mathbb{E}[\min_{i \in S} X_i] \geq \sum_{j=0}^{\log m} 2^j \prod_{i \in S} p_{i,j} / 2.$$

■

By Lemma 4.4.2, we use $\sum 2^j \prod_{i \in S} p_{i,j}$ to approximate $\mathbb{E}[\min_{i \in S} X_i]$. Now we define the signature of set S to be $\mathbf{Sg}(S) = (z_0, z_1, \dots, z_{\log m})$ with $z_j = \lfloor -\sum \log_2 p_{i,j} \rfloor$. Define $\text{Val}(\mathbf{sg}) = \sum_j 2^{j-z_j}$. It is obvious that $z_0 \leq z_1 \leq \dots \leq z_{\log m}$. The set of all possible signatures \mathbf{SG} are monotone $\log m$ -dimensional integer vectors with each coordinate at most $\log \frac{m}{\varepsilon W}$. Here are two useful facts about the signature.

Lemma 4.4.3. *For any set S ,*

$$\frac{1}{4} \text{Val}(\mathbf{Sg}(S)) \leq \mathbb{E}[\min_{i \in S} X_i] \leq \text{Val}(\mathbf{Sg}(S))$$

Proof. Let $\mathbf{Sg}(S) = (z_0, \dots, z_{\log m})$. Recall that $z_j = \lfloor -\sum_{i \in S} \log_2 p_{i,j} \rfloor$. Hence

$$z_j \leq -\sum \log_2 p_{i,j} \leq z_j + 1.$$

This simply implies

$$\sum_{i \in S} 2^j \prod p_{i,j} \leq \text{Val}(\mathbf{Sg}(S)) \leq 2 \sum_{i \in S} 2^j \prod p_{i,j}.$$

Combine with Lemma 4.4.2 we finish the proof. ■

Lemma 4.4.4. *Any coordinate of guessing value W is at least $\log \frac{1}{W} - 2$. By only consider the signature set \mathbf{SG} , the difference to optimal objective value is at most $O(\varepsilon)W$.*

Proof. Since we guess the objective value is W , by Lemma 4.4.3, $\text{Val}(\mathbf{Sg}(S)) \leq 4W$, which implies for any j , $2^{j-z_j} \leq 4W$. By a simple calculation we get $z_j \geq \log \frac{1}{W} - 2$.

On the other hand, for any signature $\mathbf{sg} = (z_0, \dots, z_{\log m})$ with $z_0 \leq z_1 \leq \dots \leq z_l \leq \log \frac{m}{\varepsilon W} < z_{l+1} \leq \dots \leq z_{\log m}$, we compare it with $\mathbf{sg}' = (z_0, \dots, z_l, \log \frac{m}{\varepsilon W}, \dots, \log \frac{m}{\varepsilon W})$. It is

obvious that $\text{Val}(\text{sg}) \leq \text{Val}(\text{sg}')$. And

$$\text{Val}(\text{sg}') - \text{Val}(\text{sg}) = \sum_{i=l+1}^{\log m} (2^{i - \log \frac{m}{\varepsilon W}} - 2^{i - z_i}) \quad (4.3)$$

$$\leq \sum_{i=l+1}^{\log m} 2^i \frac{\varepsilon W}{m} \quad (4.4)$$

$$\leq 2\varepsilon W. \quad (4.5)$$

Therefore for any signature sg not in the set SG , there is a signature $\text{sg}' \in \text{SG}$ satisfying $\text{Val}(\text{sg}) \leq \text{Val}(\text{sg}') \leq \text{Val}(\text{sg}) + 2\varepsilon W$. ■

Each signature is a $\log m$ -dimensional monotone integer vector with coordinates in the range $[\log \frac{1}{W} - 2, \log \frac{m}{\varepsilon W}]$. Hence

$$|\text{SG}| = \binom{\log \frac{m}{\varepsilon W} - \log \frac{1}{W} + 2 + \log m}{\log m} = O(m^2/\varepsilon)$$

and we can enumerate it. When getting the possible signature $z = (z_0, \dots, z_{\log m})$, let $a_{i,j} = -\log_2 p_{i,j}$ and we define the following IP:

$$\begin{aligned} \min. & \sum_i c_i x_i \\ \text{s.t.} & \sum_i a_{i,j} x_i \geq z_j & j = 0, \dots, \log m. \\ & x_i \in \{0, 1\} & i = 1, \dots, n. \end{aligned} \quad (4.6)$$

The meaning of variables and constraints in this IP is immediate:

- x_i represents the choice of X_i . $x_i = 1$ if and only if $i \in S$.
- Each constraint bounds a coordinate of $\text{Sg}(S)$ by z_j , and bound the value $\text{Val}(\text{Sg}(S)) \leq \text{Val}(\text{sg})$.

To solve IP(4.6) approximately, we propose the following theorem.

Theorem 4.4.5. *Given $\delta > 0$ and a column monotone IP*

$$\begin{aligned} \min. \quad & \sum_{i=0}^n c_i x_i \\ & Ax \geq b \\ & x \in \{0, 1\}^n. \end{aligned}$$

where b is a positive integer vector and $\frac{\max b_i}{\min b_i} \leq T$. If there is a fractional solution x^* satisfying $x^* \in [0, \theta]^n$ where $\theta = \frac{1}{2(1+\log 1/\delta)}$, there is an algorithm to find a integer solution x with probability $1 - 2\delta$ such that

$$\sum_i c_i x_i \leq O(\log \log T) \text{OPT},$$

where the constant in O notation depends on δ , and

$$Ax \geq b.$$

Proof. We round each x_i^* to 1 with probability $\frac{x_i^*}{\theta}$, and 0 otherwise. We repeat rounding by $l = \frac{\log t}{\log \frac{1}{\delta}} + 1$ turns, and set $x_i = 1$ if it is round to 1 in any round. To show its correctness, we define some things first.

We define some sets S_1, \dots, S_t according to b : $S_i = \{k \mid b_k \in (C^{i-1} \min b_i, C^i \min b_i)\}$. Here $C = \frac{1 - \sqrt{\frac{\log \frac{1}{\delta}}{1 + \log \frac{1}{\delta}}}}{\theta}$ and $t = \log_C T$. We also define a series of events $\Omega_1, \dots, \Omega_t$: $\Omega_i = \{\forall k \in S_i, \sum_j a_{j,k} x_j \geq b_k\}$.

In one round, for any $k \in S_i$, $\mathbb{E}[\sum_j a_{j,k} x_j] = \sum_j a_{j,k} \frac{x_j^*}{\theta} \geq \frac{C^{i-1}}{\theta}$. As $C\theta < 1$ and by Chernoff bound we have

$$\Pr\left[\sum_j a_{j,k} x_j < C^i\right] \leq e^{-\frac{1-C\theta}{2\theta}} = \delta.$$

$$\begin{aligned}
\Pr[\Omega_i] &= \Pr[\forall k \in S_i, \sum_j a_{j,k} x_j \geq b_k] \\
&\geq \Pr[\forall k \in S_i, \sum_j a_{j,k} x_j \geq C^i] \\
&= \Pr[\sum_j a_{j,k_0} x_j \geq C^i] \quad (k_0 \text{ is the smallest index in } S_i) \\
&= 1 - \Pr[\sum_j a_{j,k_0} x_j < C^i] \\
&\geq 1 - \delta.
\end{aligned}$$

As there are l rounds,

$$\begin{aligned}
\Pr[\forall i, \Omega_i] &\geq 1 - t\delta^l \\
&\geq 1 - \delta
\end{aligned}$$

The first inequality comes from union bound. Hence $Ax \geq b$ with probability at least $1 - \delta$.

On the other hand, $\mathbb{E}[\sum c_i x_i] \leq \frac{l}{\theta} \text{OPT}$. By Markov inequality we have

$$\Pr[\sum c_i x_i \geq \frac{l}{\delta\theta} \text{OPT}] \leq \delta.$$

Here $\frac{l}{\delta\theta} = O(\log t) = O(\log \log T)$ where the constant is related to δ . Combining two tail bounds and by union bound, we have

$$\Pr[\sum c_i x_i < O(\log \log T) \text{OPT} \wedge Ax \geq b] \geq 1 - 2\delta.$$

■

Now we solve IP(4.6). First we relax the constraint $x \in \{0, 1\}^n$ to $x \in [0, 1]^n$ and obtain its optimal solution x^* . Define $S = \{i \mid x_i^* \geq \theta\}$. For any $i \in S$, we round x^* to 1; for the rest we round x^* to 1 with probability $\frac{x_i^*}{\theta}$ for l rounds, and once it is rounded to 1, set it to 1. Here θ and l are all set same as in Theorem 4.4.5.

Theorem 4.4.6. *If there is one integer solution for IP(4.6) with cost C , our rounding method finds a solution with probability at least $1 - 2\delta$ with cost at most $(\beta_C \log \log \log m)C$ and $Ax \geq z - 1$, where β_C is a constant number related to δ .*

Proof. First, the cost of elements in S is at most θOPT and $\text{OPT} \leq C$.

After rounding the elements in S , we compute the residual of the constraints to obtain $A'x \geq b$, where A' is a submatrix of A and $b_j = z_j - \sum_{i \in S} a_{i,j}x_i$. As A' is monotone we assume b is also monotone, otherwise the non-monotone constraints must be dominated by some other constraints. Consider the set $G = \{k : b_k \geq 1\}$ and the constraint $A'_G x \geq b_G$.

If $W \geq m^{-c_0}$ for some constant c_0 ,

$$\max_{i \in G} b_i \leq \max_{i \in G} z_i \leq \log \frac{m}{\varepsilon m^{-c_0}} = (c_0 + 1) \log m + \log \frac{1}{\varepsilon},$$

and as $\min_{i \in G} b_i \geq 1$ we have $\max_{i \in G} b_i / \min_{i \in G} b_i \leq (c_0 + 1) \log m + \log \frac{1}{\varepsilon}$.

If $W < m^{-c_0}$, it leads to $\min z_i > i - \log W \geq \log \frac{1}{W}$. Assume u is the smallest one in G and w is the largest one in G . Therefore $b_u = z_u - M_u$ and $b_w = z_w - M_w$ with $M_u \leq M_w$.

And

$$\begin{aligned} \frac{b_w}{b_u} &\leq \frac{\log \frac{m}{\varepsilon W} - M_w}{\max(\log \frac{1}{W} - M_u, 1)} \\ &= \frac{\log \frac{m}{\varepsilon}}{\max(\log \frac{1}{W} - M_u, 1)} + \frac{\log \frac{1}{W} - M_w}{\max(\log \frac{1}{W} - M_u, 1)} \\ &\leq \log \frac{m}{\varepsilon} + 1 \end{aligned}$$

The first inequality follows from the upper-bound and lower-bound of z .

By Theorem 4.4.5 we know that with probability at least $1 - 2\delta$ the cost is at most $O(\log \log \log m) \text{OPT}$ and $A'_G x \geq b_G$. For those $i \notin G$, $A_i x \geq 0$ always hold, which leads to $A'_i x \geq b_i - 1$. Hence finally, there exists a constant number β_C such that the total cost is at most $(\beta_C \log \log \log m)C$ and $Ax \geq z - 1$. ■

As there are only $\log m$ inequality constraints in IP(4.6), the fractional solution of it has

Algorithm 7 Approximation for MINIMUM-ELEMENT

```
1:  $U \leftarrow \emptyset$ 
2: Enumerate  $W$  from  $b$  to  $2^N b$ 
3: for each  $W$  do
4:   compute set of signatures
5:   for each signature  $z$  do
6:     solve the corresponding IP and obtain a candidate set  $S$ 
7:     if cost of  $S \leq (\beta_C \log \log \log m)C$  and  $\mathbb{E}[\min_{i \in S} X_i] < \mathbb{E}[\min_{i \in U} X_i]$  then
8:        $U \leftarrow S$ 
9:     end if
10:  end for
11: end for
12: Return  $U$ .
```

at most $\log m$ non-integer elements. By Theorem 4.4.6,

$$\Pr[\sum c_i x_i \leq (\beta_C \log \log \log m)C \wedge Ax \geq z - 1] > 0,$$

which means there exists one. Thus we can replace the rounding by enumerating all $2^{\log m} = m$ possible values to find it.

Now we are ready to prove Theorem 4.4.1, the pseudocode can be found in Algorithm 7.

proof of Theorem 4.4.1. Assume the optimal set is S^* with objective value OPT and cost at most C . It is obvious that there exists i^* such $\text{OPT} \leq 2^{i^*} b \leq 2\text{OPT}$. When $W = 2^{i^*} b$, there exists a signature z such that S^* is a solution for its corresponding IP. By Theorem 4.4.6 we can find a solution with cost at most $(\beta_C \log \log \log m)C$ and $Ax \geq z - 1$. According to $Ax \geq z - 1$,

$$\mathbb{E}[\min_{i \in S} X_i] \leq \text{Val}(z - 1) = 2\text{Val}(z) \leq O(W) \leq O(\text{OPT}).$$

Then applying the greedy algorithm in Theorem 4.2.3, we can obtain a solution with total cost at most $(\beta_C \log \log \log m)C$. ■

Chapter 5

Conclusion

In this thesis, we propose some new approaches for approximating MAXIMUM-ELEMENT and minimization version of MAXIMUM-ELEMENT. Our approach works not only for maximization but also for minimization. Our core result is a PTAS for MAXIMUM-ELEMENT, which improves the best previous constant approximation. We also have an improved approximation for MINIMUM-ELEMENT using a new integer programming.

We have considered MAXIMUM-ELEMENT problem and MINIMUM-ELEMENT problem in the non-adaptive model. However, there are still many interesting open questions. Here I list some which I think are worth considering.

1. The most important problem is to design adaptive algorithms for either MAXIMUM-ELEMENT or MINIMUM-ELEMENT. In many scenarios, make decision adaptively is possible and it can improve performance significantly.
2. Can we design FPTAS for MAXIMUM-ELEMENT problem? If not, how to prove the hardness?
3. Is there any polynomial time constant approximation for MAX_K problem?
4. Can our result for MINIMUM-ELEMENT on the extra cost be improved to a constant factor, or even $1 + \varepsilon$ for any $0 < \varepsilon < 0.5$?

5. Take the summation of the top k elements instead of the maximum. The solution for maximum is an $O(k)$ approximation for it. Is there an approximation better than $O(k)$? Is there even a PTAS?

Bibliography

- [1] Marek Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011.
- [2] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
- [3] B.C.Dean, M.X.Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. In *Proc. 45th FOCS*, 2004.
- [4] Victor Bryant and Hazel Perfect. *Independence theory in combinatorics : an introductory account with applications to graphs and transversals*. London ; New York : Chapman and Hall, 1980. Includes index.
- [5] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
- [6] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. NIPS’16, 2016.
- [7] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3):197–206, 1955.
- [8] Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth*

International Conference on Very Large Data Bases - Volume 30, VLDB '04, pages 588–599. VLDB Endowment, 2004.

- [9] Ashish Goel, Sudipto Guha, and Kamesh Munagala. Asking the right questions: Model-driven optimization using probes. In *Proceedings of the Twenty-fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '06, pages 203–212, New York, NY, USA, 2006. ACM.
- [10] Ashish Goel, Sudipto Guha, and Kamesh Munagala. How to probe for an extreme value. *ACM Trans. Algorithms*, 7(1):12:1–12:20, December 2010.
- [11] 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.
- [12] Sudipto Guha, Kamesh Munagala, and Saswati Sarkar. Information acquisition and exploitation in multichannel wireless systems. *IEEE Transactions on Information Theory*, 2007.
- [13] Anupam Gupta, Martin Pál, R Ravi, and Amitabh Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 417–426. ACM, 2004.
- [14] Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.
- [15] Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 797–806, Washington, DC, USA, 2011. IEEE Computer Society.
- [16] Jian Li and Yu Liu. Approximation algorithms for stochastic combinatorial optimization problems. *Journal of the Operations Research Society of China*, 4(1):1–47, 2016.

- [17] Jian Li and Wen Yuan. Stochastic combinatorial optimization via poisson approximation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 971–980, New York, NY, USA, 2013. ACM.
- [18] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.
- [19] Roger Myerson and Mark A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, 1983.
- [20] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14(1):265–294, 1978.
- [21] Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *J. ACM*, 44(4):585–591, July 1997.
- [22] Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, 2006.
- [23] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

Acknowledgment

First of all, I would like to thank my mentor, Assistant Professor Li Jian, for his guidance and encouragement on my research. It is my great honor to be a student of Professor Li.

Next, I would like to thank IIS. It gives me a chance to know so many good people and chance to learn what I am interested in. Communication with others and learning helps me not only in research but also in life. It is lucky to be a member of IIS.

And I want to say thank you to my thesis reviewers, Professor Chen Wei and Professor Duan Ran. Thanks for reviewing my thesis carefully. They give me much useful advice and I benefit a lot.

Finally, thank all guys who have helped me.

Curriculum Vitae

Resume

Born on September 25,1989.

Admitted to Institute for Interdisciplinary Information Sciences, Tsinghua University in August, 2008, and got the Bachelor degree in computer science and technology in 2012.

Admitted to Institute for Interdisciplinary Information Sciences, Tsinghua University for a Master degree in August, 2012.

Publication

1. Jian Li and **Yu Liu**. Approximation Algorithms for Stochastic Combinatorial Optimization Problems. In Journal of the Operations Research Society of China, 4(1):1-47, 2016. (SCI, WOS:000383045300001)
2. Wei Chen, Wei Hu, Fu Li, Jian Li, **Yu Liu** and Pinyan Lu. Combinatorial Multi-Armed Bandit with General Reward Functions. NIPS'16, 2016.
3. Hu Ding, **Yu Liu**, Lingxiao Huang, and Jian Li. K -Means Clustering with Distributed Dimensions. ICML 2016: 1339-1348.